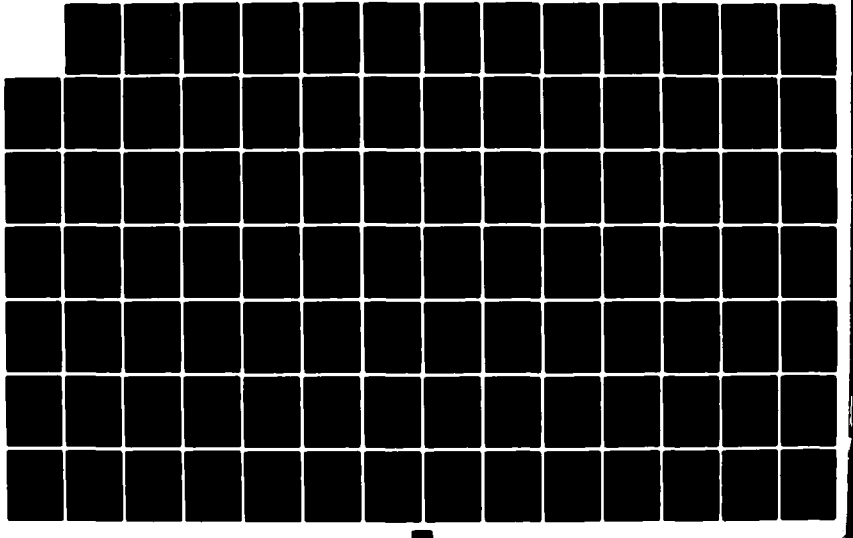
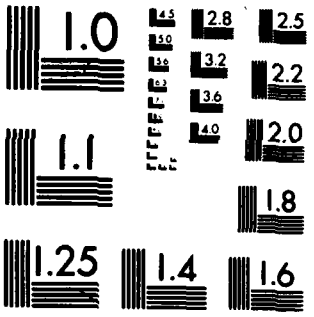
AD-A124 399 B7700 NETED REFERENCE MANUAL(U) DAVID W TAYLOR NAVAL 1/2
SHIP RESEARCH AND DEVELOPMENT CENTER BETHESDA MD
COMPUTATION MATHEMATICS/LOGISTICS DEPT
UNCLASSIFIED D V SOMMER ET AL. NOV 82 DTNSRDC/CMLD-82/27 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

CMLD-82-27

AD A124399

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Maryland 20084

B7700 NETED REFERENCE MANUAL

by
David V. Sommer
Edwin L. Zivi Jr.

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

Computation, Mathematics and Logistics Department
Departmental Report

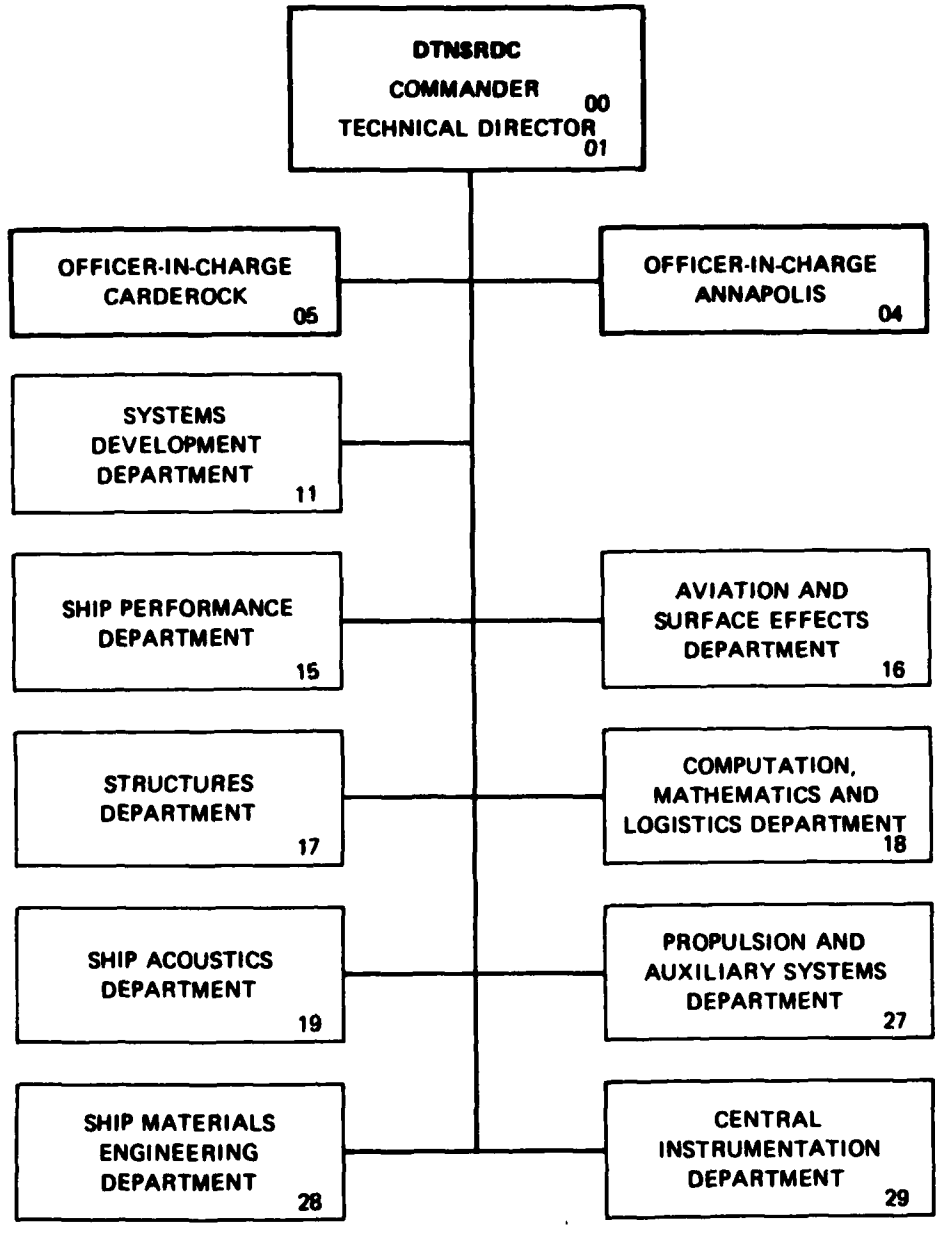
November 1982

DTNSRDC / CMLD-82-27

B7700 NETED Reference Manual

NDW-DTNSRDC 5602/30 (2-80)
(supersedes 3960/46)

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



GPO 866 987

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CMLD-82-27	2. GOVT ACCESSION NO. AD-A124399	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) B7700 NETED Reference Manual		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David V. Sommer and Edwin L. Zivi Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS DTNSRDC User Services, Code 1892 Bethesda, Maryland 20084		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Computation, Mathematics & Logistics Department, Computer Facilities Division (189)		12. REPORT DATE November 1982
		13. NUMBER OF PAGES 141
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Editor
Text Editor

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The B7700 NETED Reference Manual describes the NETED text editor, used for preparing computer programs, data files, documents and other publications. Each of the 115 NETED commands is described on a separate page, as well as in a cross-referencing summary by function. A discussion of the history of NETED as well as its many features, a sample annotated session and a list of all diagnostics messages are included.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

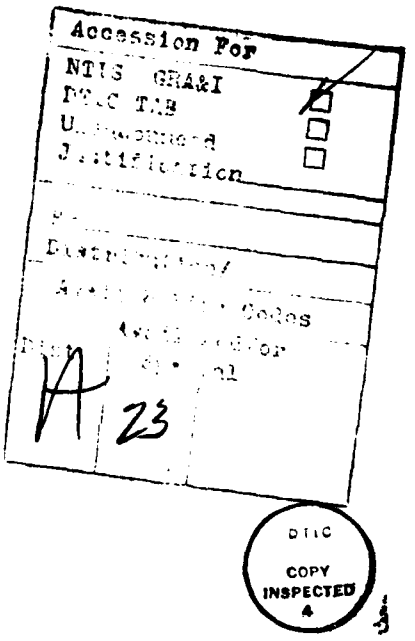
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

David W. Taylor
Naval Ship Research and Development Center
Bethesda, Maryland 20084

B7700 NETED Reference Manual

by
David V. Sommer
Edwin L. Zivi Jr.



Computation, Mathematics and Logistics Department
Departmental Report

November 1982

CMLD-82-27

Table of Contents

NETED At DTNSRDC	1-1
All Those NETED Commands	1-2
NETED and CANDE	1-6
NETED and the Operating System	1-6
Filekinds Supported	1-6
Line Numbers vs Sequence Numbers	1-7
Toggles and Switches	1-7
Strings in NETED	1-8
Interrupting NETED	1-8
Special NETED Files	1-9
Writing Workfile Lines on Disk	1-10
File Security	1-10
Compiling and Running a Program	1-11
The NETED Commands Alphabetically	2-1
A Summary of the NETED Commands by Function	3-1
Toggles and Switches	3-3
Commands Which Affect Commands	3-5
Commands Which Move The Pointer Without Changing the Workfile	3-6
Print Commands Which Never Move The Pointer	3-8
Commands Which Alter Lines	3-9
Commands Which Operate in Hexadecimal	3-11
File Manipulation Commands	3-12
Commands Which Terminate NETED	3-14
Specialized Commands	3-15
Special Listing Commands	3-16
Text/Word Processing Related Commands	3-17
Miscellaneous Commands	3-18

Appendices

Appendix A: Executing NETED A-1

Appendix B: NETED messages B-1

*** NETED At DTNSRDC ***

CDC 6000 NETED was acquired from Ed Fourt of Lawrence Berkeley Labs, implemented by the Construction Engineering Research Laboratory (CERL), Computer Services Branch, and is maintained by DTNSRDC. CDC NETED is written in Fortran IV Extended, version 4, and Compass (assembly language). It is noticeably faster and less expensive than the CDC Editor and has gained great popularity at DTNSRDC.

In 1979, after the B7700 was installed at DTNSRDC, there was a need for a pointer-oriented editor since the CANDE editor required sequence numbers. To meet this need, and to provide a common editor, CDC NETED was re-written in Algol for the B7700. As the power and ease of writing in Algol became apparent, new commands and capabilities were added and B7700 NETED now has some 115 commands, file backup, system and NETED fault recovery, and many other features. In implementing the CDC NETED commands on the B7700, the E> and I> prompting and the "*" toggle controlling it, and the NEPOS toggle were eliminated. Because a line starting with a question mark, "?", is considered a CANDE control command, the "?" option on the C, RC and UC commands was moved to the end of the command with the other options. Otherwise, all CDC NETED commands are available and work the same way on the B7700 NETED.

*** All Those NETED Commands! ***

Don't let the large number of NETED commands scare you! After you have become familiar with the basic commands, then try a few more. Eventually, you will probably settle on a group which will meet most of your editing needs.

A lot of commands have similar functions. For instance, consider the search commands: there are four types (the search string must begin in column 1 (F), in any column (L), in a specified column or column range (both equal (S) and not equal (XS) tests). For each of these 4, the search may be forward or backward (the next letter is B: FB, LB, SB, XSB). Each of these 8 will search to the next line which meets the test. By adding A to any of the eight, all lines meeting the test may be listed. Thus, we have 16 search commands: F/FA/FB/FBA, L/LA/LB/LBA, S/SA/SB/SBA, XS/XSA/XSB/XSBA.

Similarly, there are four basic ways to write the workfile to disk:

1) from top-of-file to the current line (WTOP, WDTOP);
2) the entire workfile (W, SAVE);
3) from the current line for a number of lines (WL, WDL);
4) from the current line to a line meeting a string test
   (including the line: WINS, WDINS;
   not including the line: WTOS, WDTOS).

A 'D' as the second character of the write command means the lines are deleted from the workfile after being written to disk.

The NETED commands, which may be entered in upper or lower case, are grouped below in 5 levels:

Level 0 -- basic commands
Level 1 -- starting intermediate commands
Level 2 -- intermediate commands
Level 3 -- advanced commands
Level 4 -- specialized commands

Level 0 (basic):

AR/A	Add (append) on Right
B	go to Bottom of file
C	Change one string to another
CANDE	suspend NETED to go back to CANDE
D	Delete line(s)
EDIT	EDIT another file
F	Find forward (column 1)
FB	Find Backward (column 1)
H/HELP	list the commands at the terminal
I	Insert a line
L	Locate forward (any column)
LB	Locate Backward (any column)
N	go N lines forward or backward
P	Print line(s)
QUIT	QUIT NETED, do not save edit file
R	Replace a line
RA	print RAnge of lines around current line
RC	Repeat last Change
SAVE	SAVE edit file and quit NETED
STAB	Set TAB character and tab columns
T	go to Top of file
W	Write the entire edit file
X	type a line of column markers (....5...10...)
-	go back one line (same as N -1)
<cr>	(carriage return) go to next line (same as N 1)
.	(period) switch mode (Edit-to-Input or Input-to-Edit)

Level 1 (starting intermediate):

CAPS	Set input to all CAPS
DTOP	Delete lines from top-of-file to current line
FA	Find forward All occurrences (column 1)
FBA	Find Backward All occurrences (column 1)
FOLD	FOLD lower case to upper case
LA	Locate forward All occurrences (any column)
LBA	Locate Backward All occurrences (any column)
LN	list Line Number of current line
VER/VERSION	list NETED changes
=	(equal) repeat last command (except N, -, <cr>)

Level 2 (intermediate):

AL	Add on left of line
BR	BReak line at a specific column
CO	COpy line(s) to another place in workfile (retain the original)
DUP	DUPlicate the current line
FIX	FIX line(s)
K/KL	Keep left-most columns
KR	Keep right-most columns
M	Merge a file at the pointer
MO	MOve line(s) to another place in workfile (delete the original)
PA	Print All lines single-spaced
PD	Print lines Double spaced
PDA	Print All lines Double-spaced
POP	POP toggles
Q	set/reset Query mode
RESET	RESET toggles
S	Scan forward (specified columns)
SA	Scan forward all occurrences) (specified columns)
SB	Scan Backward (specified columns)
SBA	Scan Backward All occurrences (specified columns)
SET	SET toggles
SR	Shift lines to the Right
STR	list current STRing definitions
TL	Truncate on left of line
TR	Truncate on right of line
TRUNC	set line TRUNCation criteria
UC	Undo last Change
UNFOLD	UNFOLD upper case to lower case
V	set/reset Verify toggle
WHAT	list workfile attributes
WL	Write Lines to a file
WTOP	Write lines to a file from top-of-file TO Pointer
XS	eXclusive Scan forward (specified columns)
XSA	eXclusive Scan forward all occurrences (specified columns)
XSB	eXclusive Scan Backward (specified columns)
XSBA	eXclusive Scan Backward All occurrences (specified columns)
#	set/reset show-line-numbers toggle

Level 3 (advanced):

AUTOBACKUP	set AUTOmatic file BACKUP for writes
CB	Change Backward (towards top-of-file)
COUNTS	list workfile statistics
DB	Delete Backward
DINS	Delete INcluding line meeting <String test>
DTOS	Delete up TO line meeting <String test>
NOBACKUP	set NO automatic file BACKUP for writes
PB	Print Backward (towards top-of-file)
PFN	change the workfile Permanent File Name
PH	Print line(s) in Hexadecimal
PHE	Print line(s) in Hex and Ebcdic
PHWIDTH	set PH and PHE terminal line WIDTH
PINS	Print lines INcluding line meeting <String test>
PTOS	Print lines up TO line meeting <String test>
RCB	Repeat Change Backward (towards top-of-file)
REC	edit a RECovery file
SEQ	set SEQuence numbering for writes
TEMP	edit a TEMPorary file
TI	list elapsed and remaining TImes
UCB	Undo Change Backward (towards top-of-file)
WDINS	Write-and-Delete lines to a file INcluding line meeting the <String test>
WDL	Write-and-Delete Lines to a file
WDTOP	Write-and-Delete lines to a file from top-of-file TO Pointer
WDTOS	Write-and-Delete lines to a file up TO the line meeting the <String test>
WINS	Write lines to a file INcluding the line meeting the <String test>
WTOS	Write lines to a file up TO the line meeting the <String test>

Level 4 (specialized):

CENTER	CENTER line(s)
ENTER	read from an auxiliary storage device
IN	set paragraph INdentation
LEFT	LEFT-justify line(s)
LEN	list line LENgths
LPL	List Page Lengths
LM	set Left Margin
RIGHT	RIGHT-justify line(s)
RM	set Right Margin
SLOW	set/reset SLOW print toggle
STORE	write to an auxiliary storage device
%	echo text

*** NETED and CANDE ***

NETED is actually two programs. The first, NETED, is a very short program which RUNs the second program, NETED/EDIT, the actual editor. This way, the editor is considered to be a job running outside of CANDE. It is, therefore, able to recover from most system problems, including halt/loads.

*** NETED and the Operating System (MCP) ***

Because NETED is treated as a separate job, it is charged as a priority 49 batch job, which includes the batch job overhead charge, each time NETED is executed. The NETED command 'CANDE' and the CANDE control command '?HI' let the user leave and return to a single execution of NETED any number of times.

NETED stores the entire workfile in memory and lets MCP handle memory management. However, when several people are using NETED with large files, the system may slow down because the normal memory management swapping may not be fast enough. Therefore, NETED will cause the system to swap faster as the workfile gets larger. Because of this, NETED users having large files will have slower response for commands which require "distant" lines in the workfile (lines not close to the current position of the pointer). To improve your own NETED response, work in small areas of your workfile, or better yet, keep the whole workfile small. The breaking points for the purpose of adjusting the swapping rate are 1000, 2000, and 3000 lines.

*** Filekinds Supported ***

NETED supports 19 FILEKINDs: Algol, Basic, Binder, Cdata, Cseq, Cobol, Dasdl, Data, DCAlgol, DMAlgol, Espol, Fortran (fixed and free format), Job, NDL, Newp, Pascal (Jovial), PL/I, and Seq. If any other FILEKIND is read, it is treated as CDATA (if UNITS=CHARACTERS) or DATA (if UNITS=WORDS).

*** Line Numbers vs Sequence Numbers ***

NETED is a pointer-oriented editor, unlike the CANDE editor, which is sequence-number-oriented. With the CANDE editor, each line has a sequence number with gaps left to insert additional lines. If there is insufficient room, the file must be resequenced to make room.

With NETED, each line in the workfile has a position relative to line 0 (<Top of file>). This relative position is called the line number. Lines in the workfile may be considered as numbered from 1 to n, without gaps. There is "infinite" room between lines for making insertions. As lines are added or deleted, the lines from the pointer through <Bottom of file> are "renumbered". Thus, line 476 is always the 476th line from <Top of file>, but the contents of the line depend on whether lines are added and/or deleted between <Top of file> and line 476. The pointer may be moved to any line and the contents of that line may be printed by entering just the line number. The line number of the current line can be printed by using the LN command.

For filekinds which have sequence numbers, when lines are written to a file by NETED, the first line is given a fixed sequence number, <base> (default: 1). The sequence number for each succeeding line is obtained by adding an increment, <incr> (default: 1), to the sequence number of the previous line. BASIC is discussed below. The NETED SEQ command may be used to specify a base and increment to be used when writing to a file. This base and increment have no effect on the line number within the workfile, but are used only when a file is written. To find the current base and increment, use 'SEQ ?'.

BASIC is a special case because the source program uses the sequence numbers for branching (in all other languages, the sequence numbers are ignored). BASIC programs are created by entering the sequence number followed by the statement. NETED cannot resequence a BASIC program. It can be resequenced using the CANDE editor.

*** Toggles and Switches ***

Toggles (which are either set or reset) and switches (which have one of several settings) control the operation of NETED.

One toggle (.) controls the mode (Edit or Input); one toggle (CAPS) controls input; five toggles (AUTOBACKUP/NOBACKUP, SLOW, V, #) and two switches (PHWIDTH, TRANS) control output; and one toggle (Q) controls certain NETED commands.

A toggle is changed from set-to-reset or from reset-to-set by specifying the toggle. A switch is changed by giving it a new value. Three NETED commands (SET, RESET, POP) allow one or more toggles to be changed at once.

An alphabetical list of the toggles and switches can be found on pages 3-3 and 3-4.

*** Strings in NETED ***

NETED identifies <strings> in one of two ways: In the C command, the <string>s are delimited by a character (except ? or blank) which is not in either <string>. In all other string commands (A/AR, AL, F/L/S/XS/etc., I, R), the <string> starts after the command (after allowing for at most one blank as a separator) and continues through the last character entered (trailing blanks are significant).

After a <string> has been defined for a particular string command, it may be referenced later as part or all of a <string> in any other string command by using an <esc> sequence as follows:

<esc>0 -- refers to the most recent F/L/S/XS/etc <string> (1)
<esc>1 -- refers to the most recent C <string1> (1)
<esc>2 -- refers to the most recent C <string2> (1)
<esc>3 -- refers to the most recent AL <string> (1)
<esc>4 -- refers to the most recent A/AR <string> (1)
<esc>5 -- refers to the most recent I <string> (1)
<esc>6 -- refers to the most recent R <string> (1)

*** Interrupting NETED ***

Many individual NETED commands can be interrupted while they are processing.

Any command which is printing may be terminated by hitting the <break> key. It may print a couple more lines but will end with the system message BREAK ON OUTPUT.

For disk reads (EDIT, M), disk writes (W, WL, etc.), searches (F, L, S, XS, etc.) , and changes (C, RC, UC, etc.), "? <mixno> HI" will interrupt the command, display how many lines have been processed, and ask if the user wishes to continue or not.

---

(1) At version 1.11, all <esc>n's may be used, but only in the C/CB commands.

*** Special NETED Files ***

There are several files which NETED may create or use during a given execution.

BACKUP/<filename>

When a write command is entered, lines are written to the specified or implied file <filename>. If a file already exists with that <filename>, the user has the option of replacing the existing file, renaming the existing file before writing the new file, or aborting the command.

If the user chooses to replace the existing file and AUTOBACKUP is set, the existing file is renamed as BACKUP/<filename> before the workfile lines are written. If an old BACKUP/<filename> exists, the new BACKUP/<filename> replaces the old version.

If NOBACKUP is set, the existing file is renamed as NETED/WORKFILE/TEMPORARY/<temporary number>, the workfile lines are then written and, finally, the temporary file is removed. If the write is not successful, the temporary file is *not* removed. It may be edited using the NETED command

TEMP <temporary number>

or by using "TEMP <temporary number>" as the <filename> in the RUN statement or EDIT command. After it is in the workfile, the temporary file is removed.

NETED backup files are just a means of preserving the most recently SAVEd copy of the edited file and should not be confused with B7700 "backup" print files.

NETED/WORKFILE/RECOVERY/<filename>

In most cases, when there is a fault (in NETED or the system), NETED will write the current workfile into the recovery file, NETED/WORKFILE/RECOVERY/<recovery number>, where <recovery number> is the mix number of NETED/EDIT. A recovery file may be re-edited using the NETED command

REC <recovery number>

or by using "REC <recovery number>" as the <filename> in the RUN statement or EDIT command. After it is in the workfile, the recovery file is removed.

NETED/WORKFILE/TEMPORARY/<filename>

See BACKUP/<filename>.

*** Writing Workfile Lines on Disk ***

The following options may be used for any of the write commands (SAVE/W/WDINS/WDL/WDTOP/WDTOS/WINS/WL/WTOP/WTOS/WU). A write command may be followed by a colon (:) and a write option. The following write option is supported:

:S A FILETYPE=5 file is written. The lines are squished (trailing blanks, sequence numbers (except BASIC, COBOL, CSEQDATA) and the ID field are removed). Generally, squished files require 1/3 to 1/2 the disk space, but are not compatible with CANDE or the compilers. A file which is normally saved in squished format may be unsquished by EDITing it into NETED and SAVEing it without ':S'. If this option is used on the OPEN statement, all writes to that file will be squished.

*** File Security ***

The file security attributes of existing files owned by the executing usercode are retained. New files being created and files owned by another usercode are given the default attributes of PRIVATE I/O.

*** Compiling and Running a Program ***

Since NETED is only a text editor, CANDE may be used to compile and execute programs developed using NETED.

The CANDE command

COMPILE <sourcefilename>

will compile the program in file <sourcefilename>, listing any errors at the terminal. If there are no errors, the object code file will be saved in file OBJECT/<sourcefilename>, replacing any existing file with the same name.

To run the object program, use the CANDE command

RUN <sourcefilename>

which will execute the object code in file OBJECT/<sourcefilename>.

Only the simplest forms of COMPILE and RUN are shown above. Since they are executed in CANDE, any form which CANDE accepts may be used.

The example on the next page illustrates creating a program, compiling and executing it. The following symbols are used to annotate the example:

c> -- a CANDE response
n> -- a NETED response
o> -- user program output
p> -- a CANDE prompt
u> -- a user entry
% -- descriptive comments follow a percent (%)

```
u>   RUN *NETED ("MYPROG")          % create Fortran program MYPROG
c>   #RUNNING 2740                  % NETED begins
p>   #?
n>   >RUNNING 2741                  % NETED begins
n>   97700 NETED
n>   Workfile is FORTRAN (fixed format)
n>   Truncation length =  72
n>   Tab character = \ tab stops =  7, 10, 13, 16, ...
n>   Input.                              % Input mode.  Enter program
u>    $ RESET FREE
u>   C\test SQRT
u>     100 CONTINUE
u>   \READ (5, /, END=200) A
u>   \PRINT //, "SQRT(", A, ")=", SQRT(A)
u.   \GO TO 100
u>     200 CONTINUE
u>   \STOP
u>   \END
u>   .                          % end of program, switch to Edit mode
n>   Edit.
u>   0 P $                                      % list the workfile
n>   <Top of file>
n>    $ RESET FREE
n>   C      test SQRT
n>     100 CONTINUE
n>         READ (5, /, END=200) A
n>         PRINT //, "SQRT(", A, ")=", SQRT(A)
n>         GO TO 100
n>     200 CONTINUE
n>         STOP
n>         END
n>   <Bottom of file>
u>   W                              % write workfile on file MYPROG
n>       9 lines written to (user)MYPROG.
u>   CANDE                                          % suspend NETED
n>   NETED suspended for 30 minutes (suicide at 16:14!).
     To resume: ? 2741 HI
u>   C MYPROG                                  % compile the program
c>   #COMPILING 2765
c>   FT=...
u>   R MYPROG                                    % run the program
c>   #RUNNING 2768
p>   #?
u>   25
o>   SQRT( 25.0 )= 5.0
u>   50
o>   SQRT( 50.0 )= 7.0710678118
u>   ?END                                       % terminate MYPROG
c>   #
c>   #ET=...
u>   ? 2471 HI                                     % resume NETED
p>   #?
u>   % change the program or EDIT a new file
```

*** The NETED Commands Alphabetically ***

All NETED commands have the following syntax:

[<line#>] [<toggles> :] <command> [: <options>] [<parameters>]

where [...] indicates an optional item.

<line#>
: If specified, the pointer is moved to this line before the rest of the command is processed. If a valid command fails for any reason, including a syntax error, the pointer is left at the new position.

<toggles> :
: If specified, the listed toggles are changed, for the current command only, as follows:
  <toggle> -- flip the toggle
  + <toggle> -- set the toggle
  - <toggle> -- reset the toggle
  The colon ':' is required to separate the <toggles> from the <command>.

<command>
: The command to be executed.

: <options>
: Qualifying options which control such things as filenames and formats.

<parameters>
: The presence or absence of <parameters> depends on the individual command. Generally, if the first character of <parameters> is a non-letter, then the blank separating it from <command> (or <command>:<options>) is not required.

In the examples in the rest of the chapter, each line is preceded by one of:

```
 u>          -- a user entry
-->          -- the position of the pointer after successful
                completion of the command
<nothing>    -- typed output of the command
```

A (Append)
AL (Add on Left)
AR (Add on Right)

(Append) A
(Add on Left) AL
(Add on Right) AR

AR <string> or A <string>
AR or A

AL <string>
AL

Append <string> to the end of the current line (A/AR) or add at the start of the line (AL). If <string> is omitted, the most recent corresponding definition is used.

See page 1-9: Strings in NETED.

See K/KL/KR, SR, STR, TL/TR.

Toggles affecting the output of this *command*: V, N.

Pointer after command completion: unchanged.

Examples:

```
 u> P
--> The quick brown fox jumped
 u> A  over
--> The quick brown fox jumped over
 u> F The gam
--> The game is
 u> A
--> The game is over

 u> L rain
--> over the rainbow
 u> AL Somewhere          % including one space at the end
--> Somewhere over the rainbow
```

AUTOBACKUP
NOBACKUP

AUTOBACKUP
NOBACKUP

AUTOBACKUP

NOBACKUP

When AUTOBACKUP is set, each write command which writes to an existing file will rename the existing file to BACKUP/<filename> before writing the new file. If BACKUP/<filename> already exists, it is purged.

When NOBACKUP is set, BACKUP/<filename> is not created. Instead, an existing file is renamed NETED/WORKFILE/TEMPORARY/<temporary number> until the new file is written successfully. Then the temporary file is removed.

Default: AUTOBACKUP.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

1)
```
u> W MYFILE
   File exists, rename old version or <cr>.
u> <cr>
   Backup up.
      50 lines written to MYFILE.
```

2)
```
u> NOBACKUP
   NOBACKUP set.
u> W MYFILE
   File exists, rename old version or <cr>.
u> <cr>
      50 lines written to MYFILE.
```

B (Bottom of file) (Bottom of file) B

B

Move the pointer to the last line of the file and print the line.

Toggles affecting the output of this command: V, #.

Pointer after command completion: the last line of the file.

Example:

```
 u> 1 P -$
--> line 1
    line 2
    ...
    line last
    <Bottom of file>
 u> B
--> line last
```

BR (BReak a line) (BReak a line) BR

BR <col>

Break the current line at column <col>. Original columns 1 - <col>-1 will become one line; columns <col> - $ will become a new line. The pointer will be positioned at the new line.

<col> must be greater than 1 and less than or equal to the current line length (or '$').

Toggles affecting the output of this command: V, #.

Pointer after command completion: the new line.

Example:

```
 u> P
--> This is a line to be broken.
 u> X 33
    ....5...10...15...20...25...30...
 u> BR 15
--->  to be broken.
 u> RA
-1- This is a line
--->  to be broken.
+1+ The next line.
```

C	(Change)	(Change)	C
CB	(Change Backward)	(Change Backward)	CB
RC	(Repeat Change)	(Repeat Change)	RC
RCB	(Repeat Change Backward)	(Repeat Change Backward)	RCB
UC	(Undo Change)	(Undo Change)	UC
UCB	(Undo Change Backward)	(Undo Change Backward)	UCB

```
C <dlm><string1><dlm><string2><dlm> (<cols>) <nlines> *<ntimes>
                                    <global> <?>
CB <dlm><string1><dlm><string2><dlm> (<cols>) <nlines> *<ntimes>
                                     <global> <?>

RC (<cols>) <nlines> *<ntimes> <global> <?>
RCB (<cols>) <nlines> *<ntimes> <global> <?>

UC (<cols>) <nlines> *<ntimes> <global> <?>
UCB (<cols>) <nlines> *<ntimes> <global> <?>
```

Change <string1> to <string2> in a forward (C/RC/UC) or backward (CB/RCB/UCB) direction.

<dlm> is a delimiter and may be any character not in either string, except a blank.

If <cols> is specified, it has one of the following forms:

<col1>
: <string1> must start in column 1.

<col1>-<col2>
: <string1> must be contained within <col1> thru <col2>.

<col1>:<ncols>
: <string1> must be contained within <col1> thru <col1>+<ncols>-1.

<col2> or <ncols> may be $ to indicate end-of-line.

<nlines> is the number of lines to be changed. It changes the current line and the next (or next preceding) <nlines>-1 lines. If <nlines> is omitted, only the current line is changed. If <nlines> is preceded by a minus sign '-' sign, the pointer is returned, after changing, to its original position.

'* <ntimes>' causes the first <ntimes> occurrences of <string1> to be changed.

If <global> is the letter G, all occurrences of <string1> are changed on each of the <nlines> lines. If <global> is omitted, only the first occurrence of <string1> is changed (in each of the <nlines> lines).

If <?> is a question mark '?', each line changed will be typed and one of the following responses must be given:

```
Y   -- make the change
C   -- make the change and continue without question
N   -- no change, but proceed with the command
Q   -- quit the command now, without change
```

If <string1> is null (<dlm><dlm>), then <string2> is placed in front of the first (or all, if <global> is specified) column in the specified or implied <cols>. This is the same as AL <string2>.

The options (<cols>), <nlines>, *<ntimes>, <global>, and <?>, may be given in any order. Any or all may be omitted.

Hint: If the options are omitted and the final <dlm> is forgotten, RC will execute the command, eliminating the need to re-type it.

'<esc>1' may be used to refer to the most recent "change" <string1> as part or all of any <string> in any string command.

'<esc>2' may be used to refer to the most recent "change" <string2> as part or all of any <string> in any string command.

See page 1-9: Strings in NETED.

RC/RCB repeats the last command.

UC/UCB undoes the last command. It is shorthand for
```
   C /<esc>2/<esc>1/...
   CB /<esc>2/<esc>1/...
```
but does not change the values of <esc>1 and <esc>2.

See STR.

Toggles affecting the output of this command: V, #.

	Show-line-numbers on	Show-line-numbers off
Verify set	changed lines listed with line number	changed lines listed without line number
Verify reset	line numbers of changed lines listed; asterisk indicates multiple changes	only error and truncation messages

Pointer after command completion:
If <nlines>: C/RC/UC : current line + <nlines> - 1.
CB/RCB/UCB : current line - <nlines> + 1.
If - <nlines>: unchanged.

Examples:

1)
```
u> P
--> Corect this line
u> C /r/rr/
--> Correct this line
```

2)
```
u> SET #
u> P -3
--> 7 This line has a error.
    8 This is ok
    9 Here's aother error.
u> -V : C / a/ an/ -3
    7 9
u> V
    Verify reset.
u> -# : P 3
    This line has an error.
    This is ok
--> Here's another error.
```

3)
```
u> P
--> the first shall be last
u> C /la/fir/
--> the first shall be first
u> UC
--> the last shall be first
```

CANDE CANDE

CANDE <minutes>

Suspend NETED for <minutes> minutes (default: 30) and allow execution of CANDE commands. Note that CANDE control commands (those starting with ?) may always be entered at any time and in either mode (Input or Edit).

In response to CANDE, NETED will type

```
NETED suspended for <minutes> minutes (suicide <time>!)
To resume: ? <mixno> HI
```

When ready to resume NETED, enter

```
? <mixno> HI
```

where <mixno> is the mix number for NETED. If forgotten, as may be the case when using a CRT, use

```
?J          -or-          ?MIX
```

to obtain the mix number.

If NETED is not resumed within the specified or implied time, it will create a recovery file and terminate without notifying the terminal.

While NETED is suspended, use of the CANDE commands BYE, HELLO or SPLIT will either terminate the session or initiate a new session with a new mix number after which the suspended NETED may not be restarted.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Example:

```
u> ?TIME
   #12:27 PM TUESDAY, JUNE 22, 1982
u> CANDE 45
   NETED suspended for 45 minutes (suicide at 10:40!)
   To resume: ? 4753 HI
u> (any number of CANDE commands)
u> ? 4753 HI
   #?
u> (next NETED command)
```

CAPS (input CAPS lock toggle) (input CAPS lock toggle) CAPS

CAPS

Flip the toggle controlling upper/lower case on lines entered from the keyboard.

When CAPS is set, all lower case input is translated to upper case.

When reset, no translation takes place, that is, lower case letters are kept in lower case.

See FOLD/UNFOLD.

Default: reset.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> CAPS
   Caps set.
u> I Now is the time ...
u> P
--> NOW IS THE TIME ...

u> CAPS
   Caps reset.
u> I The quick brown fox ...
u> P
--> The quick brown fox ...
```

CENTER (CENTER within margins) (CENTER within margins) CENTER
LEFT (LEFT-justify) (LEFT-justify) LEFT
RIGHT (RIGHT-justify) (RIGHT-justify) RIGHT

CENTER <nlines>
LEFT <nlines>
RIGHT <nlines>

Center, left- or right-justify <nlines> lines within the current margins.

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is 0 or omitted, only the current line is changed. If <nlines> is preceded by a minus sign '-', the pointer is returned, after the change, to its original position.

If a line is longer than the current margin width, it is not changed.

See LM/RM.

Toggles affecting the output of this command: V, #.

Pointer after command completion:
If <nlines>: current line + <nlines> - 1.
If - <nlines>: unchanged.

Examples:

```
u> P -3
   First line to be processed
   second line
   line 3
u> LM ?
   LM=1  RM=30  IN=0  width=30
u> X 30
   ....5....10....15....20....25....30
u> RIGHT -3
-->     First line to be processed
                        second line
                             line 3
u> CENTER -3
-->   First line to be processed
                second line
                   line 3
u> LEFT3
--> First line to be processed
   second line
   line 3
```

CO (COpy) (COpy) CO

<line#> CO <amount to copy> <destination>

Copy lines to the specified place.

<amount to copy> is <nlines> -- a specific number of lines

<destination> is one of:

AT <line#>	-- after line <line#>
+ <nlines>	-- after current line + <nlines>
- <nlines>	-- after current line - <nlines>

If the lines have been copied successfully, one of the messages:

<n> lines copied, inserted after line <m>.
<n> lines copied, inserted after <top-of-file>
<n> lines copied, inserted at <bottom-of-file>

is typed.

See DUP, MO.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
 u> P -4
--> <Top of file>
    line 1
    line 2
    line 3
    <Bottom of file>
 u> 1 CO 2 + 1
    2 lines copied, inserted after line 2.
 u> P 5
    line 1
    line 2
    line 1
    line 2
--> line 3
 u> CO $ AT 0
    1 lines copied, inserted after <Top-of-file>.
 u> 0 P $
    <Top of file>
    line 3
    line 1
    line 2
    line 1
    line 2
    line 3
--> <Bottom of file>
```

COUNTS (list workfile statistics) (list workfile statistics) COUNTS

COUNTS

List the following statistics about the workfile:

number of lines
number of characters
length of the shortest line
length of the longest line

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> COUNTS
   275 lines, 13759 characters
   Shortest =   7, Longest =  72.
```

D (Delete) (Delete) D
DB (Delete Backward) (Delete Backward) DB

D <nlines>

DB <nlines>

Delete <nlines> lines forward (D) or backward (DB), starting with the current line.

If <nlines> is omitted, only the current line is deleted.

See DTOP, DTOS/DINS.

Toggles affecting the output of this command: V.

Pointer after command completion: the line before the first (or only) line deleted.

Examples:

```
u> RA 2
   -2- line d
   -1- line e
--> >>> line f
   +1+ line g
   +2+ line h
u> D 2
u> RA
   -1- line d
--> >>> line e
   +1+ line h
```

```
u> RA 2
   -2- line d
   -1- line e
--> >>> line f
   +1+ line g
   +2+ line h
u> DB 2
u> RA
   -1- line c
--> >>> line d
   +1+ line g
```

DTOP (Delete TO Pointer) (Delete TO Pointer) DTOP

DTOP

Delete all lines from <Top of file> up to, but not including, the current line.

See D/DB, DTOS/DINS.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Example:

```
u> RA 4
   -4- <Top of file>
   -3- line 1
   -2- line 2
   -1- line 3
--> >>> line 4
   +1+ line 5
   +2+ line 6
   +3+ line 7
   +4+ line 8
u> DTOP
u> RA 4
   -1- <Top of file>
--> >>> line 4
   +1+ line 5
   +2+ line 6
   +3+ line 7
   +4+ line 8
```

DTOS (Delete TO String) (Delete TO String) DTOS
DINS (Delete INcluding String) (Delete INcluding String) DINS

DTOS <string test>

DINS <string test>

Delete all lines from the current line through and including (DINS) or not including (DTOS) the first line which satisfies the <string test>.

The <string test> has one of the following forms:
F <string>
L <string>
S (<cols>) <string>
XS (<cols>) <string>
which have the same meanings as when used alone as commands.

If query is set, the line satisfying the <string test> is typed. If this is the desired terminating line, respond Y or C; if not, respond N or Q.

If no line is found which satisfies the <string test>, the command is not executed.

See D/DB, DTOP.

Toggles affecting the output of this command: Q, V, #.

Pointer after command completion: current line - 1.

Examples:

```
 u> BA 3                          u> BA 3
    -1- <Top of file>                -1- <Top of file>
--> >>> line 1                   --> >>> line
    +1+ line 2                       +1+ line 2
    +2+ line 3                       +2+ line 3
    +3+ line 4                       +3+ line 4
 u> DINS S (6) 3                  u> DTOS S (6) 3
    line 3                           line 3
    ?>                               ?>
 u>    Y                          u>    Y
    3 lines deleted.                 2 lines deleted.
 u> P 2                           u> P 3
    <Top of file>                    <Top of file>
--> line 4                           line 3
                                 --> line 4
```

DUP (DUPlicate) (DUPlicate) DUP

DUP <ntimes>

Duplicate the current line <ntimes> times following the current line. Except for the pointer position after the command, this is the same as "CO 1 + 0" done <ntimes> times.

If <nlines> is preceded by a minus sign (-), the pointer is not moved.

See CO, MO.

Toggles affecting the output of this command: V.

Pointer after command completion:
if <nlines>: at the <ntimes>th copy.
if - <nlines>: unchanged.

Examples:

```
u> RA
   -1- line p
--> >>> line q
   +1+ line r
u> DUP
u> RA 2
   -2- line p
   -1- line q
--> >>> line q
   +1+ line r
   +2+ line s

u> RA
   -1- line j
--> >>> line k
   +1+ line l
u> DUP 3
u> RA 4
   -4- line j
   -3- line k
   -2- line k
   -1- line k
--> >>> line k
   +1+ line l
   +2+ line m
   +3+ line n
   +4+ line o
```

EDIT (EDIT a new workfile) (EDIT a new workfile) EDIT

```
EDIT "<filename>,<filekind>,<length>"
EDIT "-<filename>,<filekind>,<length>"
```

The NETED workfile is cleared and re-initialized. The file <filename> is edited into the workfile.

The parameters are the same as in the RUN statement.

The quotes ("...") are optional.

If <filename> is preceded by a minus sign '-', only the filekind, maximum line length and number of lines (if <filename> already exists) are listed.

If the current workfile has not been saved, NETED will respond with:

```
Workfile not saved.
?>
```

Enter 'Y' or 'C' to continue, or 'N' or 'Q' to abandon the EDIT command and regain the current workfile.

Toggles affecting the output of this command: V.

Pointer after command completion: at <top-of-file>.

Example:

```
 u> RA
    -1- <Top of file>
--> >>> line 1
    +1+ line 2
 u> EDIT OLD/FILE
    Workfile is FORTRAN (fixed format)
    Maximum line length =  72
     1278 lines read from OLD/FILE.
    Edit.
 u> P
--> <Top of file>
 u> $
--> <Bottom of file>
```

```
u> EDIT NEW/FILE,CO
   Workfile is COBOL
   Truncation length =  66
   Tab character = \ tab stops = 4, 8, 12, 16, ...
   Input.
u> .
   Edit.
u> B
--> <Top of file>

u> EDIT -NEXTFILE
   ALGOL tr=72 6235 lines.
   Edit.
```

ENTER (ENTER from remote storage) (ENTER from remote storage) ENTER

ENTER
ENTER:S <nsecs>

Lines are input from an auxiliary storage device which uses the standard DC1-DC4 codes to provide remote control of read/write functions.

This command causes NETED to switch to the input mode and send a DC1 ("start read") to the terminal to initiate reading from the storage device. (If a line with "bit" is typed and the command does not end, press the delete key (DEL or <ctrl>-X).

If ':S' is specified, no 'DC' codes are sent to the terminal.

The ENTER command terminates when no input has for <nsecs> seconds. The default is 5 seconds.

Since execution of this command is limited by the terminal's baud rate and may take quite a while, the final line rings the bell five times to signal completion.

After completion of this command, NETED is returned to the edit mode.

This command is useful for transferring files from the CDC 6000 to the B7700.

See STORE.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

The floppy disk has a DC3 "stop read" code:

```
u> ENTER
   Input.
      <lines from the floppy disk are listed>
     117 lines entered.
   Edit.
```

The floppy does not have a DC3 "stop read" code:

```
u> ENTER
   Input.
      <lines from the floppy disk are listed.
       When the last desired line has been read/listed,
       press the READ button on the 9512.>
      47 lines entered.
   Check the last line(s).
   Edit.
```

```
F   (Find)                                                    (Find)  F
FB  (Find Backward)                                 (Find Backward) FB

L   (Locate)                                                (Locate)  L
LB  (Locate Backward)                            (Locate Backward) LB

S   (Scan)                                                    (Scan)  S
SB  (Scan Backward)                                 (Scan Backward) SB

XS  (eXclusive Scan)                                  (eXclusive Scan)  XS
XSB (eXclusive Scan Backward)              (eXclusive Scan Backward) XSB
```

```
F <string>
FB <string>

L <string>
LB <string>

S (<cols>) <string>
SB (<cols>) <string>

XS (<cols>) <string>
XSB (<cols>) <string>
```

Forward (backward) search for the next (next previous) line containing (F,FB,L,LB,S,SB) or not containing (XS,XSB) the <string>. They differ in where the test is made in the line:

```
F/FB   --  <string> must start in column 1

L/LB   --  <string> may be anywhere in the line

S/SB   --  <string> must start in <col1> (if (<col1>)
           <string> must start with <col1>-(<col1>+<ncols>-1)
                    (if (<col1>:<ncols>))
           <string> must be within <col1>-<col2>
                    (if (<col1>-<col2>))

XS/XSB --  <string> must not start in <col1> (if (<col1>)
           <string> must not be within <col1>-<col2>
                    (if (<col1>-<col2>))
           <string> must start with <col1>-(<col1>+<ncols>-1)
                    (if (<col1>:<ncols>))
```

See page 1-9: Strings in NETED.

See STR.

Toggles affecting the output of this command: V, #.

	Show-line-numbers on	Show-line-numbers off
Verify set	each line is listed with line number	each line is listed without line number
Verify reset	only line numbers are listed	no output

Pointer after command completion:
if found: at the line containing the <string>.
if not found: unchanged.

Examples:

```
u> RA 4
   -4- Now is the
   -3- time for
   -2- all good men
   -1- to come to the
--> >>> aid of
   +1+ the quick brown
   +2+ fox who jumped
   +3+ over the lazy
   +4+ dog.
 u> F the
--> the quick brown
 u> FB ti
--> time for
 u> L women
   Not found.
 u> P
--> time for
 u> LB w
--> Now is the
 u> S (6) the
--> over the lazy
 u> SR (2-4) w
--> Now is the
```

```
FA   (Find All)                                          (Find All)  FA
FBA  (Find Backward All)                        (Find Backward All) FBA

LA   (Locate All)                                        (Locate All)  LA
LBA  (Locate Backward All)                      (Locate Backward All) LBA

SA   (Scan All)                                          (Scan All)  SA
SBA  (Scan Backward All)                        (Scan Backward All) SBA

XSA  (eXclusive Scan All)                       (eXclusive Scan All)  XSA
XSBA (eXclusive Scan Backward All)   (eXclusive Scan Backward All) XSBA


     FA <string>
     FBA <string>

     LA <string>
     LBA <string>

     SA (<cols>) <string>
     SBA (<cols>) <string>

     XSA (<cols>) <string>
     XSBA (<cols>) <string>
```

Forward (or backward) search for next (or next previous) lines through <bottom of file> (or <top of file>) for all lines containing (FA/FBA/LA/LBA/SA/SBA) or not containing (XSA/XSBA) the <string>. They differ in where the test is made in the line:

```
      FA/FBA   --  <string> must start in column 1

      LA/LBA   --  <string> may be anywhere in the line

      SA/SBA   --  <string> must start in <col1> (if (<col1>))
                   <string> must start with <col1>-(<col1>+<ncols>-1)
                            (if (<col1>:<ncols>))
                   <string> must be within <col1>-<col2>
                            (if (<col1>-<col2>))

     XSA/XSBA  --  <string> must not start in <col1> (if (<col1>))
                   <string> must start with <col1>-(<col1>+<ncols>-1)
                            (if (<col1>:<ncols>))
                   <string> must not be within <col1>-<col2>
                            (if (<col1>-<col2>))
```

See page 1-9: Strings in NETED.

See STR.

Toggles affecting the output of this command: V, #.

	Show-line-numbers on	Show-line-numbers off
Verify set	each line is listed with line number	each line is listed without line number
Verify reset	only line numbers are listed. For LA/LBA, an asterisk before a line number indicates a multiple occurrence of the <string>	

Pointer after command completion: unchanged.

Examples:

```
u> WHAT
   ...
   Workfile is FORTRAN
   ...
u> FA C
   (All comments in a Fortran program)
u> 0 XSA (1) C
   (All Fortran program statements except comments)
```

FIX (FIX a line) (FIX a line) FIX

```
FIX
FIX <nlines>
FIX <
FIX <<
FIX >
FIX >>
```

Make one or more changes to a line.

For the first two forms, the line to be fixed will be typed. The user then enters a line composed of a combination of the following editing characters or break phrases(trailing blanks may be omitted):

Editing characters:

```
<space>  --  retain the above character
   B     --  change the above character to a blank
   D     --  delete the above character
   F     --  fold the above character
   I     --  insert text before the above character
   R     --  replace the above character
   U     --  unfold the above character
```

Break phrases (single-line FIXes only):

(For the following 5, <text> means from the start of the current line thru and including the marked character)

```
     <  -- move <text> to a new line preceding the current line
   n  <  -- move <text> to the previous line before column n
   n << -- move <text> to the previous line before column n
           leaving a blank before the moved text
   $  <  -- move <text> to the end of the previous line
   $ << -- move <text> to the end of the previous line
           leaving a blank before the moved text
```

(For the following 5, <text> means from the marked character thru the end of the line)

```
   >     -- move <text> to a new line following the current line
   >  n -- move <text> to the next line before column n
   >> n -- move <text> to the next line before column n
         and insert a blank after the moved text
   >  $ -- move <text> to the end of the next line
   >> $ -- same as '> $'
```

B/D/F/U editing is done first. Then the modified line is typed up to the first 'I' or 'R'. The user enters the new text. This is repeated until all 'I' and 'R' editing has been completed. Note that a field of R's is treated as if it were a field of D's followed by 'I'.

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is changed. If <nlines> is preceded by a minus sign, '-', the pointer is returned, after the changes, to its original position. Note that changes are indicated only for the first line. The same changes are made to all <nlines> lines.

Break editing ('<', '>') is valid only for single line fixes, that is, when <nlines> is 1.

Four special forms of FIX will concatenate lines:

FIX <

Concatenate the current line and the preceding line. That is, add the current line to the end of the preceding line and delete the current line. This has the same effect as using '$>' at the end of the current line and then deleting the current (now blank) line.

FIX <<

Same as 'FIX <', except that a space is inserted between the two concatenated texts.

FIX >

Concatenate the current line and the next line. That is, add the current line to the start of the next line and delete the current line. This has the same effect as using '>0' at the start of the current line and then deleting the current (now blank) line.

FIX >>

Same as 'FIX >', except that a space is inserted between the two concatenated texts.

Toggles affecting the output of this command: V, #.

Pointer after command completion:
If <nlines>: current line + <nlines> - 1.
If - <nlines>: unchanged.

Examples:

1)
```
u> FIX
          a234556 ==7.3 * G
u>        F   D    BI      I
          A23456 =
u>                  ZZ + (
          A23456 = ZZ + (7.3 * G
u>                           )
          A23456 = ZZ + (7.3 * G)
```

2)
```
u> 168 FIX
   This is a line to be fixxed and broken up.
u>                           D        >
   This is a line to be fixed and
--> broken up.
```

3)
```
 u> 168 P -2
--> This is a line to be fixxed and broken up.
    This is the next line.
 u> FIX
--> This is a line to be fixxed and broken up.
 u>                          D        >0
    This is a line to be fixed and
--> broken up.This is the next line.
```

4)
```
 u> 168 P -2
--> This is a line to be fixxed and broken up.
    This is the next line.
 u> FIX
--> This is a line to be fixxed and broken up.
 u>                          D        >17
    This is a line to be fixed and
--> This is the next broken up.line.
```

5)
```
 u> 168 FIX
    This is a line to be fixxed and broken up.
 u>                          D        <
--> This is a line to be fixed and
    broken up.
```

```
6)      u> 168 RA
        u> FIX
       --> This is a line to be fixxed and broken up.
        u>                              D       $<
       --> The previous line.This is a line to be fixed and
           broken up.
        u> 168 RA
           -1- The previous line.
       --> >>> This is a line to be fixxed and broken up.
           +1+ This is the next line.
        u> FIX
       --> This is a line to be fixxed and broken up.
        u>                              D       12<
       --> The previousThis is a line to be fixed and line.
           broken up.

7)      u> 473 # : RA 2
            -2- line 471
            -1- line 472
           473> line 473
            +1+ line 474
            +2+ line 475
        u> FIX <
       --> line 472line 473
        u> RA
           -1- line 471
       --> >>> line 472line 473
           +1+ line 474

8)      u> 473 # : RA 2
            -2- line 471
            -1- line 472
           473> line 473
            +1+ line 474
            +2+ line 475
        u> FIX >>
       --> line 473 line 474
        u> RA
           -1- line 472
       --> >>> line 473 line 474
           +1+ line 475
```

FOLD
UNFOLD

FOLD
UNFOLD

FOLD (<cols>) <nlines>

UNFOLD (<cols>) <nlines>

Change all lower case letters to upper case (FOLD) or all upper case letters to lower case (UNFOLD) for <nlines> starting with the current line.

If <cols> is specified, it has one of the following forms:

<col1>
Only one column is FOLDed/UNFOLDed

<col1>-<col2>
All columns between <col1> and <col2>, inclusive, are FOLDed/UNFOLDed

<col1>:<ncols>
<ncols> beginning with column <col1> are FOLDed/UNFOLDed

<col2> or <ncols> may be $ to indicate end-of-line.

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is changed. If <nlines> is preceded by a minus sign, '-', the pointer is returned, after the change, to its original position.

See CAPS.

Toggles affecting the output of this command: V, #.

Pointer after command completion:
If <nlines>: current line + <nlines> - 1.
If - <nlines>: unchanged.

Example:

```
 u> RA 3
    -1- <Top of file>
--> >>> AbCdEfGhIjKlMnOpQrStUvWxYz01253467890()+-...
    +1+ aBcDeFgHiJkLmNoPqRsTuVwXyZ01235467890()+-...
    +2+ ALL UPPER CASE
    +3+ all lower case
 u> 2 FOLD 3
    ABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890()+-...
    ALL UPPER CASE
--> ALL LOWER CASE
 u> -
--> ALL UPPER CASE
 u> UNFOLD (5-9)
--> ALL upper CASE
 u> 0 P 5
    <Top of file>
    AbCdEfGhIjKlMnOpQrStUvWxYz01234567890()+-...
    ABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890()+-...
    ALL upper CASE
--> ALL LOWER CASE
```

HELP (user HELPs) (user HELPs) HELP

HELP	or	H
HELP <type>	or	H <type>
HELP <command>	or	H <command>

HELP/H types the following menu of HELP lists available:

```
NETED HELP MENU:
 H    : This menu
 H 1  : Toggles and switches
 H 2  : Commands which affect commands
 H 3  : Commands which move the pointer but do not change the
        workfile
 H 4  : Print commands which never move the pointer
 H 5  : Commands which alter lines
 H 6  : Commands which operate in hexadecimal
 H 7  : File manipulation commands
 H 8  : Commands which terminate or suspend NETED
 H 9  : Text/word processing related commands
 H 10 : Commands which write to the printer
 H 11 : Specialized commands
 H 12 : Miscellaneous commands
 H 13 : List of commands having individual HELPs
 H <-1 or any negative number> : H 1 thru H 13, inclusive
 H <1 or any positive number gt 13>  : Alphabetical list of all
                                       commands (toggles and
                                       switches separate)
```

HELP/H <type> types the requested list(s).

HELP/H <command> types a summary of the specified <command>. Use H 13 for a list of <command>s having individual HELPs.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Example:

```
u> HELP 6
   NETED <version> file manipulation commands:
    ...
   ----- End of HELP 6 -----

u> HELP $
   NETED <version> commands:
    ...
   ----- End of HELP $ -----

u> HELP FIX
   FIX command summary:
    ...
   ----- End of HELP FIX -----
```

I (Insert) (Insert) I

I <string>

Insert a new line after the current line.

See page 1-9: Strings in NETED.

See STR.

Toggles affecting the output of this command: none.

Pointer after command completion: at the inserted line.

Example:

```
 u> P -3
--> line q
    line r
    line s
 u> I new line q1
 u> RA
    -1- line q
--> >>> new line q1
    +1+ line r
```

IN (set paragraph INdentation) (set paragraph INdentation) IN

IN <ncols>
IN ?

Set paragraph indentation, relative to LM, for use by future commands FILL, and FILLJ.

<ncols> may be positive or negative, subject to the restriction that:

-(LM-1) <= <ncols> <= (RM-LM)

If <ncols> is $ or -$, the indentation is defined as the largest or smallest value, respectively. That is, (RM-LM) or -(LM-1).

If <ncols> is omitted, the paragraph indentation is reset to 0.

"IN ?" will display the current left and right margins, paragraph indentation and line width.

See LM/RM.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> IN ?
   LM=10  RM=72  IN=0  width=63

u> IN 5
   LM=10  RM=72  IN=5  width=63

u> IN -$
   LM=10  RM=72  IN=-9  width=63
```

K (Keep left-most columns) (Keep left-most columns) K
KL (Keep Left-most columns) (Keep Left-most columns) KL
KR (Keep Right-most columns) (Keep Right-most columns) KR

K <ncols> <nlines>
KL <ncols> <nlines>

KR <ncols> <nlines>

Shorten <nlines> lines by keeping columns on the left (K/KL) or right (KR) and throwing away the rest of each line. These are the same as: TR <line length-ncols> <nlines> and TL <line length-ncols> <nlines>.

<nlines> is the number of lines to be truncated. It truncates the current line and the next <nlines>-1 lines. If <nlines> is omitted, only current line is truncated. If <nlines> is preceded by a minus sign, '-', the pointer is returned, after the change, to its original position.

See A/AR/AL, SR, TL/TR.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
 u> P
--> This line was too long (by ? characters)
 u> X
    ....5...10...15...20...25...30...35...40...45...50
 u> K 22
--> This line was too long

 u> P
--> This line was too long (by ? characters)
 u> KR 22
--> long (by ? characters)
```

LEN (list line LENgth) (list line LENgth) LEN

LEN <nlines> <long>

List the length of each of <nlines> lines starting with the current line.

If <nlines> is omitted, only the length of the current line is listed. If <nlines> is preceded by a minus sign "-", the pointer is returned, after listing, to its original position.

If <long> is specified, lines longer than <long> characters will be flagged with an asterisk and a count of the excess characters.

Toggles affecting the output of this command: #.

Pointer after command completion:
if <nlines>: current line + <nlines> - 1
if - <nlines>: unchanged

Examples:

```
u> X 30
   ....5...10...15...20...25...30
u> P
--> this line has 23 chars.
u> LEN
   23
u> LEN 1 10
   23 * 11
```

LM (set Left Margin)
RM (set Right Margin)

(set Left Margin) LM
(set Right Margin) RM

```
LM <lm>
LM ?

RM <rm>
RM ?
```

Set the left and right margin for CENTER, LEFT and RIGHT commands.

<lm> must be positive and less than <rm>. (Default: 1)

<rm> must be positive, greater than <lm> and less than the maximum line length. (Default: the maximum line length)

'LM ?' and 'RM ?' will display the current left and right margins, paragraph indentation, and line width.

See IN.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> LM ?
   LM=1  RM=80  IN=0  width=80
u> LM 7
   LM=7  RM=80  IN=0  width=74
u> RM 66
   LM=7  RM=66  IN=0  width=60
u> LM
   LM=1  RM=66  IN=0  width=66
u> RM
   LM=1  RM=80  IN=0  width=80
```

LN (Line Number) (Line Number) LN

LN

List the line number of the current line.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Example:

```
u> RA 2
   -2- <Top of file>
   -1- line 1
--> >>> line 2
   +1+ line 3
   +2+ line 4
u> LN
   Line      2
```

LPL (List Page Lengths) (List Page Lengths) LPL

LPL <page length>

List page lengths for the document currently in the workfile. The following carriage control characters in column 1 are recognized:

1	- top of page
space	- print on the next line
0	- double space
-	- triple space (note that this is not recognized by the 87700 printer)
+	- print on the same line

Any other character in column 1 will be ignored.

Pages greater than <page length> lines are flagged with an asterisk (*) and a count of the number of extra lines in the page.

A summary of the carriage control characters encountered is printed at the end.

If <page length> is preceded by a minus sign, individual page lengths and the locations of bad CC's are not listed. Only the total line and CC summary are printed.

If verify is set, the bad carriage control characters, if any, are listed.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> LPL
   Pg # lines * >60
    1 60
    2 54
    3 27
    4 65 * 5
    ...
   15 60
   Total: 15 pages, 724 lines, (3 too long)

   CC summary:
   '1'  148
   ' '  3062
   '0' 1660
   '+'  147
   'S'  150
   bad    4
```

M (Merge file) (Merge file) M

```
M <filename>                              (1)
M <filename> <line#>                      (2)
M <filename> <from> - <to>                 (3)
M <filename> <from> - $                    (4)
M <filename> <from> : <nlines>             (5)
M <filename> <from> : $                    (6)
```

Merge lines from file <filename> after the current line.

M has six forms:

(1) merge a complete file
(2) merge one line
(3) merge a range of lines
(4) merge a range of lines through <Bottom of file>
(5) merge a certain number of lines
(6) merge a range of lines through <Bottom of file>
(same as (4))

Toggles affecting the output of this command: V.

Pointer after command completion:

if any lines merged: at the last (or only) line merged into the workfile.
if no lines merged: unchanged.

Example:

```
u> CANDE
   NETED suspended for 30 minutes (suicide at 9:13!).
   To resume: ? 1234 HI
u> LIST MRG : U
   #FILE (user)MRG ON DTNSRDC
   Line M1
   Line M2
   Line M3
   #
u> ? 1234 HI
   #?
u> RA $
   -2- <Top of file>
   -1- Line n-1
--> >>> Line n
   +1+ Line n+1
   +2+ <Bottom of file>
u> M MRG
        3 lines read from (user)MRG.
u> RA
   -1- Line M2
--> >>> Line M3
   +1+ Line n+1
u> $ M MRG 1
        1 lines merged from (user)MRG.
u> RA
   -1- Line n+1
--> >>> Line M1
   +1+ <Bottom of file>
```

MO (MOve) (MOve) MO

<line#> MO <amount to move> <destination>

Move lines to the specified place, deleting the original lines.

<amount to move> is <nlines> -- a specific number of lines

If preceded by a minus sign (-), the pointer points to the first of the moved lines.

<destination> is one of:

TO <line#>	-- after line <line#>
+ <nlines>	-- after current line + <nlines>
- <nlines>	-- after current line - <nlines

If the lines have been moved successfully, one of the messages:

<n> lines moved, inserted after line <m>.
<n> lines moved, inserted after <top-of-file>
<n> lines moved, inserted at <bottom-of-file>

See CO, DUP.

Toggles affecting the output of this command: V.

Pointer after command completion:
if <nlines>: at the last line moved.
if - <nlines>: at the first line moved (in its new position).

Examples:

```
u> P 5
   <Top of file>
   Line 1
   Line 2
   Line 3
--> Line 4
 u> 2 MO 1 + 1
   1 lines moved, inserted after Line 2 (updated).
 u> 0 P 5
   <Top of file>
   Line 1
   Line 3
   Line 2
--> Line 4
 u> 1 MO -2 TO $
   2 lines moved, inserted at <Bottom-of-file>.
 u> RA $
   -1- <Top of file>
   -2- Line 2
   -1- Line 4
--> >>> Line 1
   +1+ Line 3
   +2+ <Bottom of file>
```

N (go to Next-th line)
<cr> (go to next line)
- (go back 1 line)

(go to Next-th line) N
(go to next line) <cr>
(go back 1 line) -

N <nlines>
<cr>
-

Move the pointer foward <nlines> lines.

If <nlines> is omitted, the pointer is moved to the next line.

Two short forms are available to move forward (<cr>) or backward (-) one line.

Toggles affecting the output of this command: V, #.

Pointer after command completion:
N : at current line + <nlines>
<cr>: at current line + 1
- : at current line - 1

Examples:

```
u> #
   Show line numbers set.
u> P - 2
--> 26 line 26
   line 27
u> <cr>
--> 27 line 27
u> -
--> 26 line 26
u> N 10
--> 36 line 36
```

P (Print)
PB (Print Backward)
PD (Print Double-spaced)
<line#>

(Print) P
(Print Backward) PB
(Print Double-spaced) PD
<line#>

P <nlines> (<cols>)
PB <nlines> (<cols>)
PD <nlines> (<cols>)
<line#>

Print <nlines> lines forward (P/PD/<line#>) or backward (PB) starting with the current line. Use P/PB for single-spaced list; PD for double-spaced list.

<nlines> is the number of lines to be printed. It prints the current line and the next (or next preceding) <nlines>-1 lines. If <nlines> is omitted, only the current line is printed. If <nlines> is preceded by a minus sign, '-', the pointer is returned, after printing, to its original position.

If (<cols>) is specified, only those columns are printed. If not specified, the entire line is printed.

'<line#>' is the same as '<line#> P 1'.

Toggles affecting the output of this command: #.

Pointer after command completion:
if <nlines>: P/PD : current line + <nlines> - 1.
PB : current line - <nlines> + 1.
if - <nlines>: unchanged.
if just <line#> : at line <line#>.

Examples:

Print 4 lines:

```
u> P 4
   line n
   line n+1
   line n+2
--> line n+3
```

```
u> P -4
--> line n
   line n+1
   line n+2
   line n+3
```

Print 4 lines starting at line 417:

```
u> 417 P 4
   line 417
   line 418
   line 419
--> line 420
```

```
u> 417 P -4
--> line 417
   line 418
   line 419
   line 420
```

Print columns 3-7 of the lines starting at line 417:

```
u> 417 P 5 (3-7)
   ne 41
   ne 41
   ne 41
   ne 42
--> ne 42
```

```
u> 417 P -5 (3:5)
--> ne 41
    ne 41
    ne 41
    ne 42
    ne 42
```

Print entire file from any place in the file:

```
u> 0 P $
   <Top of file>
   line 1
   ...
   last line
--> <Bottom of file>
```

```
u> 0 P -$
--> <Top of file>
    line 1
    ...
    last line
      <Bottom of file>
```

Print line 1736:

```
u> 1736
--> this is line 1736
```

PA (Print All) (Print All) PA
PDA (Print Double-spaced All) (Print Double-spaced All) PDA

PA (<cols>)
PDA (<cols>)

Print all lines single-spaced (PA) or double-spaced (PDA). <Top of file> and <Bottom of file> are not printed.

If (<cols>) is specified, only those columns are printed. If not specified, the entire line is printed.

Toggles affecting the output of this command: #.

Pointer after command completion: unchanged.

Examples:

```
 u> P
--> This is line number 15
 u> PA
    <first line>
    <second line>
    ...
    <last line>
 u> P
--> This is line number 15
```

PFN (change Permanent File Name) (change Permanent File Name) PFN

PFN <filename>
PFN ?

Change the name of the workfile to <filename>.

If <filename> already exists and has different attributes (MAXRECSIZE and BLOCKSIZE), an error message is typed.

If there is a syntax error in <filename>, an error message will be printed if the Verify toggle is set.

'PFN ?' will display the current permanent file name.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Example:

```
u> PFN MY/NEW/FILE
   Warning - attributes differ.

u> PFN ANOTHER/NAME
u> PFN ?
   (user)ANOTHER/NAME.
```

PH (Print in Hexadecimal) (Print in Hexadecimal) PH
PHE (Print in Hex and Ebcdic) (Print in Hex and Ebcdic) PHE

PH <nlines> (<cols>)
PHE <nlines> (<cols>)

Print <nlines> lines in hexadecimal (PH) or EBCDIC and hexadecimal (PHE).

<nlines> is the number of lines to be printed. It prints the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is printed. If <nlines> is preceded by a minus sign, "-", the pointer is returned, after printing, to its original position.

If (<cols>) is specified, only those columns are printed. If not specified, the entire line is printed.

Printing is controlled by the PHWIDTH setting (line width). Long lines are broken at a multiple of 10 characters.

See PHWIDTH.

Toggles affecting the output of this command: V, #.

Pointer after command completion:
If <nlines>: current line + <nlines> - 1.
If - <nlines>: unchanged.

Examples:

Print 2 lines in EBCDIC and hex:

```
u> PHE -2
--> line n
    989849 444444 444444 ...
    395505 000000 000000 ...

    line n +1
    989849 4F4444 444444 ...
    395505 E10000 000000 ...
```

Print 2 lines in hex:

```
u> PH 2
    989849 444444 444444 ...
    395505 000000 000000 ...

--> 989849 4F4444 444444 ...
    395505 E10000 000000 ...
```

PHWIDTH (set PH/PHE terminal WIDTH) (set PH/PHE terminal WIDTH) PHWIDTH

```
PHWIDTH <nchar>
PHWIDTH
PHWIDTH ?
```

Define the terminal line width in characters for the PH and PHE commands. <nchar> must be greater than 0.

'PHWIDTH' will reset the terminal width to 72.

'PHWIDTH ?' will display the current setting.

Default: 72.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> PHWIDTH ?
   Terminal width = 72
u> PHWIDTH 80
   Terminal width =  80
```

POP (POP a toggle) (POP a toggle) POP

POP <toggle list>

Change the setting of each listed toggle to its next previous setting.

At most 46 settings of a toggle, including the current setting, are retained. When all stacked settings are POPped, the toggle will be reset.

<toggle list> is one or more of the toggles (except period '.'). If an option apoears more than once, each is processed and stacked for later access by POP.

See RESET, SET.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> SET V #
   % Verify set.      ------------------------------
   % Show line numbers set.      -------------      :
u> RESET V                                    :     :
   % Verify reset.      -----------------     :     :
u> SET V                                :     :     :
   % Verify set.                        :     :     :
u> POP V                                :     :     :
   % Verify reset.      -----------------     :     :
u> POP # V                                    :     :
   % Show line numbers set.      -------------      :
   % Verify set.      ------------------------------
```

PTOS (Print TO String) (Print TO String) PTOS
PINS (Print INcluding String) (Print INcluding String) PINS

PTOS <string test>
PTOS (<cols>) <string test>

PINS <string test>
PINS (<cols>) <string test>

Print all lines starting with the current line through and including (PINS) or not including (PTOS) the first line satisfying the <string test>.

If (<cols>) is specified, only those columns are printed. If not specified, the entire line is printed.

Toggles affecting the output of this command: V, #.

Pointer after command completion: at the last line printed.

Examples:

```
 u> P                          u> P
--> <Top of file>             --> <Top of file>

 u> PINS F line 3              u> PTOS F line 3
    <Top of file>                 <Top of file>
    line 1                        line 1
    line 2                    --> line 2
--> line 3

 u> PINS (6) F line 3          u> PTOS (6) F line 3
    <Top of file>                 <Top of file>
    1                             1
    2                         --> 2
--> 3
```

Q (Query toggle) (Query toggle) Q

Q

Flip the toggle which controls the 'TO <string>' commands (DINS/DTOS/WINS/WTOS/WDINS/WDTOS, but not PINS/PTOS).

When the query toggle is set, NETED will print the line satisfying the <string test> and prompt with '>?'. The user must respond with:

Y or C -- to continue the command
N or Q -- to abort the command

The Query toggle also affects the VERSION command. When the toggle is reset, NETED does not print the line or ask permission to complete the command.

When VERSION is executed with Query set, the user is asked if earlier VERSION updates are desired. With Query reset, all VERSION updates are listed.

Default: set.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> Q
   Query reset.
u> Q
   Query set.
```

QUIT QUIT

QUIT

Terminate NETED immediately. The workfile is not saved.

QUIT is not valid when working in the alternate workfile.

If the current workfile has not been saved, NETED will respond with:

```
Workfile not saved.
            --> This is line number 15
```

Enter 'Y' or 'C' to continue, or 'N' or 'Q' to abandon the EDIT command and regain the current workfile.

See DIR, SAVE.

Toggles affecting the output of this command: none.

Pointer after command completion: does not apply.

Example:

```
u> QUIT
   >ET=<et>  PT=<pt>  IO=<io>
```

R (Replace) (Replace) R

R <string>

Replace the current line.

See page 1-9: Strings in NETED.

See STR.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Example:

```
 u> P
--> the current line
 u> R A new line.
 u> P
--> A new line.
```

RA (RAnge) (RAnge) RA

RA <nlines>

List the current line and <nlines> lines on either side. Lines in front are prefixed with '-'; lines after are prefixed with '+'. The current line is prefixed with '>>>'. If <nlines> is omitted, one line on either side is listed.

Toggles affecting the output of this command: #.

Show-line-numbers toggle:
set : line number of current line is shown.
reset: current line preceded by '>>>'.

Pointer after command completion: unchanged.

Examples:

1)
```
u> RA
   -1- line n-1
--> >>> current line
   +1+ line n+1
```

2)
```
u> SET #
u> LN
   Line    49
u> RA 2
   -2- line n-2
   -1- line n-1
   49> current line
   +1+ line n+1
   +2+ line n+2
```

REC (edit a RECovery file) (edit a RECovery file) REC

REC <recovery number>
REC ?

If NETED/WORKFILE/RECOVERY/<recovery number> exists, the NETED workfile re-initialized. The specified recovery file is edited into the workfile and the recovery file is removed.

If the recovery file does not exist, the current workfile remains unchanged.

'REC ?' will list any recovery or temporary files.

See TEMP.

Toggles affecting the output of this command: V.

Pointer after command completion: at <top-of-file>.

Examples:

```
 u> RA
    -1- <Top of file>
--> >>> line 1
    +1+ line 2
 u> REC ?
    Neted recovery file(s) present:
    REC 970
 u> REC 970
    <same as for EDIT "REC 970">
 u> P
--> <Top of file>
 u> REC ?
    No recovery files.
```

RESET (RESET a toggle) (RESET a toggle) RESET

RESET <toggle list>

Reset each of the toggles in the <toggle list>. The previous setting, up to a total of 46 settings, is retained for each toggle.

<toggle list> is one or more of the toggles (except period '.'). If an option appears more than once, each is processed and stacked for later access by POP.

See POP, SET.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> RESET V #
   Verify reset.
   Show line numbers reset.
```

SAVE (SAVE workfile, exit NETED) (SAVE workfile, exit NETED) SAVE

SAVE <filename>
SAVE

The workfile is saved as <filename> and NETED is terminated. If <filename> is omitted, the <filename> is the first of:
1) the <filename> from the most recent EDIT;
2) the <filename> from the RUN *NETED statement.
If <filename> was a recovery or temporary file, the original <filename> is used.

If the file belonged to the user and had non-default security information and <filename> was omitted, that security information will be inherited by the new file. In all other cases, NETED will save files with the active default security (Private (I/O).

If <filename> already exists, the message

File exists, rename old version or <cr>.

is typed. Enter one of:
1) <cr> to replace the existing file;
2) a new <filename> to be used to rename the existing file before saving the workfile;
3) . (or any invalid <filename>) to terminate the command.

When the workfile is saved, the following message is typed:

n lines written to <filename>.

If 'SAVE:S' is specified, the lines are squished.

SAVE is not valid when working in the alternate workfile.

Note: SAVE is the same as W followed by QUIT.

Toggles affecting the output of this command: AUTOBACKUP, V.

Pointer after command completion:
if successful : does not apply.
if not successful: unchanged.

Examples:

1)
```
u> SECURITY MYFILE PUBLIC IN
#
u> RUN *NETED ("MYFILE")
...
u> SAVE
File exists, rename old version or <cr>.
u> <cr>
Backed up.
Security = public (in).
 1475 lines written to MYFILE.
>ET=<et>  PT=<pt>  IO=<io>
```

2)
```
u> SECURITY MYFILE PRIVATE IO
u> RUN *NETED ("MYFILE")
...
u> SAVE MYNEWFILE
Backed up.
 1475 lines written to MYNEWFILE.
>ET=<et>  PT=<pt>  IO=<io>
```

3)
```
u> RUN *NETED ("MYFILE")
...
u> SAVE XYZ
File exists, rename old version or <cr>.
u> XYZ/OLD
XYZ changed to XYZ/OLD
Backed up.
 1475 lines written to XYZ.
>ET=<et>  PT=<pt>  IO=<io>
```

4)
```
u> RUN *NETED ("X")
...
u> SAVE
% inherits security
```

4)
```
u> RUN *NETED ("X")
...
u> SAVE X
% uses default security
```

SEQ (SEQuencing) (SEQuencing) SEQ

```
SEQ <base> + <incr>
SEQ
SEQ ?
```

Change the output line sequencing. <base> is the starting sequence number used for the first line written. <incr> is the increment. Thus, the second line written will have sequence number <base>+<incr>; the third line, <base>+2*<incr>; etc.

SEQ affects the sequence numbers, if any, for the write commands only (SAVE/W/WDL/WDTOP/WDTOS/WL/WTOP/WTOS).

SEQ is not valid for BASIC, CDATA or DATA files.

'SEQ' restores the default (same as SEQ 1 + 1).

'SEQ ?' will display the current <base> and <incr>.

Default: 1+1.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> SEQ ?
   base + incr = 1 + 1
u> SEQ 100+100
   base + incr = 100 + 100
u> SEQ
u> SEQ ?
   base + incr = 1 + 1
```

SET (SET a toggle) (SET a toggle) SET

SET <toggle list>

Set each of the toggles in the <toggle list>. The previous setting, up to a total of 45 settings, is retained for each toggle.

<toggle list> is one or more of the toggles (except period '.').
If an option appears more than once, each is processed and stacked for later access by POP.

See POP, RESET.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> SET V #
```

SLOW (SLOW/fast print toggle) (SLOW/fast print toggle) SLOW

SLOW

When this toggle is set, NETED will prevent backspaces from being used instead of carriage returns, that is, every line will be printed from left to right. This allows lines to be printed properly on terminals which cannot backspace.

When SLOW is reset, printing is done in the normal system manner.

SLOW affects all print (or implied print) commands and is independent of the system <ctrl-z> toggle.

Default: reset.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> SLOW
   Slow printing set.

u> SLOW
   Slow printing reset.
```

SR (Shift Right) (Shift Right) SR

SR <ncols> <nlines>

Shift lines to the right by adding <ncols> blanks at the start of each line.

<nlines> is the number of lines to be shifted. It shifts the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is truncated. If <nlines> is preceded by a minus sign, '-', the pointer is returned, after the change, to its original position.

This is shorthand for C //<ncols blanks>/ <nlines>; or for the group: LM <ncols>+1; LEFT <nlines>; LM.

See A/AL/AR, K/KL/KR, TL/TR.

Toggles affecting the output of this command: V.

Pointer after command completion:
if <nlines>: current line + <nlines> - 1.
if - <nlines>: unchanged.

Examples:

```
 u> P
--> Line to be shifted.
 u> SR 5
-->      Line to be shifted.
```

STAB (Set TAB) (Set TAB) STAB

```
STAB tabch tab1, tab2, ...
STAB tabch base + incr
STAB tabch tab1, tab2, ..., base + incr
STAB tabch
STAB
STAB ?
```

Define the tab character (tabch) and up to 30 tab columns. The tab character may be any character except space and question mark (?).

1) a list of up to 30 tab columns, comma-separated. Each tabi must be less than tabj for i<j.
2) base + incr. Thirty tabs will be defined at base, base+incr, base+2*incr, ..., base+29*incr.
3) one or more comma-separated tab columns followed by a single base+incr.

Pre-defined tabs (base+incr) and maximum line lengths are as follows:

ALGOL	- 3+2	(72 characters)	
BASIC	- 5+2	(72 characters)	
BINDER	- no tabs	(72 characters)	
CDATA	- no tabs	(80 characters)	
COBOL	- 4+4	(66 characters)	
CSEQ	- no tabs	(74 characters)	
DASDL	- 3+2	(72 characters)	
DATA	- no tabs	(80 characters)	
DCALGOL	- 3+2	(72 characters)	
DMALGOL	- 3+2	(72 characters)	
FSPOL	- 3+2	(72 characters)	
FORTRAN	- 7+3	(72 characters)	(fixed format)
FREE	- 3+2	(66 characters)	(free format Fortran)
JOB	- 3+2	(80 characters)	
NDL	- 3+2	(72 characters)	
NEWP	- 3+2	(72 characters)	
PASCAL	- 3+2	(72 characters)	
PLI	- 3+2	(72 characters)	
SEQ	- no tabs	(72 characters)	

'STAB' by itself will clear the tab character and tab columns.

If the tabs have been cleared, 'STAB tabch' without any tab settings, will restore the tabs to the most recent definition. Otherwise, 'STAB tabch' changes the tab character.

'STAB ?' will list the tab character and tab columns.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> STAB ^ 11,21,31,41,46,51,56,61
u> STAB ?
   Tab character = ^ tab stops = 11, 21, 31, 41,
    46, 51, 56, 61, 64, 67, 70, 73, 76, 79, 82, 85, 88,
    91, 94, 97,100,103,106,109,112,115,118,121,124,127.

u> STAB \ 7*7
u> STAB ?
   Tab character = \ tab stops = 7, 14, 21, 28,
    35, 42, 49, 56, 63, 70, 77, 84, 91, 98,105,112,119,
    126,133,140,147,154,161,168,175,182,189,196,203,210.

u> STAB
u> STAB ?
   No tabs set.
```

STORE (STORE on remote storage) (STORE on remote storage) STORE

STORE
STORE:S <nlines>

Write lines to an auxiliary storage device, such as a floppy disk or cassette tape.

If <nlines> is omitted, the entire file is written.

STORE send a DC2 ("start write"), then the workfile lines (each followed by LF CR, then DC3 ("stop read") and DC4 ("stop write").

If ':S' is specified, no 'DC' codes are sent to the terminal and the 'lines stored' messages is not typed.

This command is known to work for the following:

1) Techtran 9512 floppy disk drive connected between an Omron terminal and the modem.

One use of this command is to transfer files from the B7700 to the CDC 6000/Cyber 74.

Since execution of this command is limited by the terminal's baud rate and may take quite a while, the final line rings the bell five times to signal completion.

See ENTER.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> -V:WHAT
   FORTRAN 642 lines.
u> STORE
      <lines written to the storage device are listed>
   642 lines stored.
```

STR (list current STRings) (list current STRings) STR

STR

List the current definitions of the strings which can be remembered by NETED: A/AR <string>, AL <string>, C/<string1>/<string2>, F <string> (L, S, XS, and their variants).

The list, which will show the <esc> sequence which may be used to refer to any of the strings, has the following format:

Currently-defined strings:

command	<esc>	length	<string>
F/L/S/XS	0	nnn	<string from most recent F/L/S/XS/FA/LA/SA/XSA command>
C 1	1	nnn	<string1 from most recent C command>
C 2	2	nnn	<string2 from most recent C command>
A/AR	3	nnn	<string from most recent A/AR command>
AL	4	nnn	<string from most recent AL command>
I	5	nnn	<string from most recent I command>
R	6	nnn	<string from most recent R command>

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> STR
   Currently-defined strings:
     command  <esc>  length  <string>
     F/L/S/XS    0      22   the last search string
     C 1         1       9   abcdefghi
     C 2         2       0
     A/AR        3      19   the <string test>.
     AL          4       2   of
     I           5      25   This is an inserted line.
     R           6      19   A replacement line.
```

T (Top of file) (Top of file) T

T

Move the pointer to the top of the file.

Toggles affecting the output of this command: V.

Pointer after command completion: at <Top of file>.
Example:

```
 u> RA
    -1- previous line
--> >>> current line
    +1+ next line
 u> T
--> <Top of file>
```

TEMP (edit a TEMPorary file) (edit a TEMPorary file) TEMP

TEMP <temporary number>
TEMP ?

The NETED workfile is cleared and re-initialized. The specified temporary file is edited into the workfile and the temporary file is removed.

"TEMP ?" will list any temporary files.

Toggles affecting the output of this command: V.

Pointer after command completion: at <top-of-file>.

Examples:

```
 u> RA
    -1- <Top of file>
--> >>> line 1
    +1+ line 2
 u> TEMP ?
    TEMP 1265
 u> -V : TEMP 1265
    <same as for EDIT "TEMP 1265">
 u> P
--> <Top of file>
 u> TEMP ?
    No temporary files.
```

TI

Show elapsed, processor and IO times used and the processor time remaining.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Example:

```
u> TI
   >ET=<et>  PT=<pt>  IO=<io>
   >Remaining PT=<pt>
```

TL (Truncate columns on the Left) (Truncate columns on the Left) TL
TR (Truncate columns on the Right) (Truncate columns on the Tight) TR

TL <ncols> <nlines>

TR <ncols> <nlines>

Shorten <nlines> lines by truncating columns on the left (TL) or right (TR).

<nlines> is the number of lines to be truncated. It truncates the current line and the next <nlines>-1 lines. If <nlines> is omitted, only current line is truncated. If <nlines> is preceded by a minus sign, '-', the pointer is returned, after the change, to its original position.

See A/AR/AL, K/KL/KR, SR.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

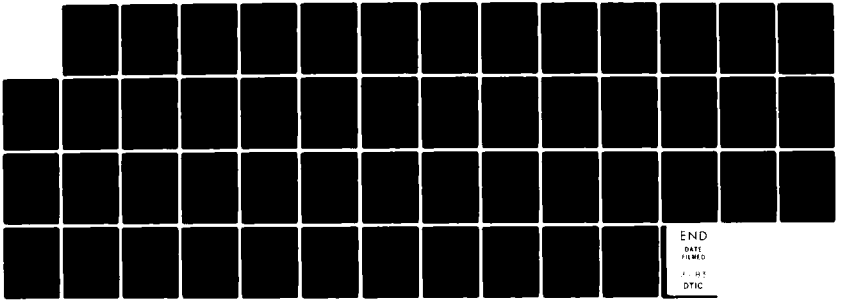
```
 u> P
--> This line was too long (by 18 characters)
 u> TR 18
--> This line was too long

 u> P
--> This line was too long (by 18 characters)
 u> TL 18
--> long (by 18 characters)
```

AD-A124 399 B7700 NETED REFERENCE MANUAL(U) DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER BETHESDA MD COMPUTATION MATHEMATICS/LOGISTICS DEPT

UNCLASSIFIED D V SOMMER ET AL. NOV 82 DTNSRDC/CMLD-82/27 F/G 9/2 NL

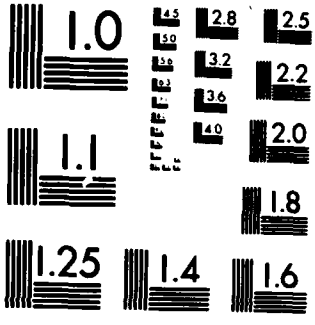
2/2



END

DATE FILMED

DTIC



TRANS (change TRANSlate table) (change TRANSlate table) TRANS

TRANS <tbl>
TRANS ?

Change the translate table used for printing at the terminal.

<tbl> is one of the following:
0 - EBCDIC to ASCII (no EOT)
(All characters, except EOT, will be transmitted without change.)
1 - non-graphic to ?
(All non-printing characters will be changed to ? before printing.)
2 - non-graphic to blank
(All non-printing characters will be changed to blank before printing.)
omitted - same as 0

'TRANS ?' will show the current table.

See PH, PHE.

Default at RUN time: 1.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> TRANS ?
   TTY Output translation = non-graphic to ?.
   Translate table menu:
     0 = full ASCII
     1 = non-graphic to ?
     2 = non-graphic to blank.

u> TRANS
   TTY Output translation = EBCDIC to ASCII.
```

V (Verify toggle) (Verify toggle) V

V

Flip the verify toggle switch.

When the verify toggle is set, all generated NETED lines are typed.

When the verify toggle is reset, lines relating to changes, pointer movement and file reading and writing are not typed. Search-all commands will list only the line numbers.

Error messages are always typed.

Default: set.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
 u> P
--> line 1

 u> V
    Verify reset.
 u> N+5
 u> P
--> line 6

 u> V
    Verify set.
 u> N+5
--> line 11
```

VERSION
VER

List the compile date and time for the current version of NETED. Also list any modifications to NETED.

If query is set, "More?" is typed and the user must respond with:
Y or C -- print the next most recent version description
N or Q -- terminate the command
This continues until N or Q is entered or all descriptions have been typed.

If query is reset, all previous version descriptions are listed without question. "END OF VERSION" signals the end of the list.

VER lists the information for just the current version of NETED.

Toggles affecting the output of this command: Q.

Pointer after command completion: unchanged.

Example:

```
u> VERSION
   NETED VERSION <version> compiled on <date> @ <time>

   ADDITIONS:
      •••

   CHANGES:
      •••

   PATCHES (corrected bugs):
      •••

   BUGS
        No known bugs.
   More ?
u>       y
   NETED VERSION <version> compiled on <date> @ <time>
      •••
   More ?
u>       y
      •••
   ==========     END OF VERSION     ==========
```

W (Write) (Write) W

W <filename>

The entire workfile is saved as file <filename>. If <filename> is omitted, the <filename> is the first of:

1) the <filename> from the most recent EDIT;
2) the <filename> from the RUN *NETED statement.

If <filename> was a recovery or temporary file, the original <filename> is used.

If the file belonged to the user and had non-default security information and <filename> was omitted, that security information will be inherited by the new file. In all other cases, NETED will save files with the active default security (Private (I/O).

If <filename> already exists, the message

File exists, rename old version or <cr>.

is typed. Enter one of:

- <cr> to replace the existing file;
- a new <filename> to be used to rename the existing file before saving the workfile;
- . (or any other invalid <filename>) to terminate the command.

When the workfile is saved, the following message is typed:

n lines written to <filename>.

If the file belonged to the user and had non-default security information and <filename> was omitted, that security information will be inherited by the new file. In all other cases, NETED will save files files with the active default security (Private (I/O).

If 'W:S' is specified, the file is squished.

Toggles affecting the output of this command: AUTOBACKUP, V.

Pointer after command completion: unchanged.

Examples:

```
u> RUN *NETED ("MYFILE")
   ...
u> W
   File exists, rename old version or <cr>.
u> <cr>
   Backed up.
    1475 lines written to MYFILE.

u> RUN *NETED ("MYFILE")
   ...
u> W MYNEWFILE
    1475 lines written to MYNEWFILE.

u> RUN *NETED ("MYFILE")
   ...
u> W XYZ
   File exists, rename old version or <cr>.
u> XYZ/OLD
   XYZ changed to XYZ/OLD
    1475 lines written to XYZ.
```

WHAT

List information about the workfile. If there are any directives in the alternate workfile, a line count is also given.

If verify is reset, a short form of the information is typed.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged.

Examples:

```
u> WHAT
   (user) MYFILE
   Workfile is COBOL
   Truncation length =  72
   Current number of lines = 1492
   TTY output translation = non-graphic to ?.
   QUERY reset, VERIFY set, show line #'s set, AUTOBACKUP
   Maxrecsize = 14 words
   Security = private (I/O)
u> -V : WHAT
   (user) MYFILE
   COBOL tr=72 1492 records
   Set: V, # AUTOBACKUP

u> WHA.
   (user) MYFILE2
   Workfile = CDATA
   Truncation length =  80
   Current number of lines = 263
   TTY output translation = non-graphic to ?.
   Current number of directives = 4
   QUERY set, VERIFY set, show line #'s reset, NOBACKUP
   Maxrecsize = 80 characters
   Security = public (in).
u> -V : WHAT
   (user) MYFILE2
   CDATA tr=80 263 lines
   4 directives
   Set: Q V.
```

WL (Write Lines) (Write Lines) WL
WDL (Write & Delete Lines) (Write & Delete Lines) WDL

WL <nlines> <filename>

WDL <nlines> <filename>

Write (WL) or write-and-delete (WDL) <nlines> lines, starting with the current line, to file <filename>.

<nlines> is required.

If <filename> already exists, the message

File exists, rename old version or <cr>.

is typed. Enter one of:

- <cr> to replace the existing file;
- a new <filename> to be used to rename the existing file before saving the workfile;
- . (or any other invalid <filename>) to terminate the command.

When the workfile is saved, the following message is typed:

n lines written to <filename>.

If 'WL:S' or 'WDL:S' is specified, the lines are squished.

Toggles affecting the output of this command: AUTOBACKUP, V.

Pointer after command completion:
WL : unchanged.
WDL: current line - 1.

Examples:

```
u> WL 26 FILE1
      26 line written to FILE1.

u> 147 WL:S 130 FILE2
   File exists, rename old version or <cr>.
u> <cr>
   File squished.
     130 lines written to FILE2.
```

WTOP (Write TO Pointer) (Write TO Pointer) WTOP
WDTOP (Write & Delete TO Pointer) (Write & Delete TO Pointer) WDTOP

WTOP <filename>
WDTOP <filename>

Write (WTOP) or write-and-delete (WDTOP) all lines from the top of the file up to, but not including, the current line into file <filename>.

If <filename> already exists, the message

File exists, rename old version or <cr>.

is typed. Enter one of:

- <cr> to replace the existing file;
- a new <filename> to be used to rename the existing file before saving the workfile;
- . (or any other invalid <filename>) to terminate the command.

When the workfile is saved, the following message is typed:

n lines written to <filename>.

If 'WTOP:S' or 'WDTOP:S'' is specified, the lines are squished.

Toggles affecting the output of this command: AUTOBACKUP, V.

Pointer after command completion: unchanged.

Examples:

```
u> LN
   Line    47
u> WTOP FILE3
      46 lines written to FILE3.

u> 924 WTOP FILE4
     923 lines written to FILE4.
```

WTOS (Write TO String) (Write TO String) WTOS
WDTOS (Write & Delete TO String) (Write & Delete TO String) WDTOS

WINS (Write INcluding String) (Write INcluding String) WINS
WDINS (Write & Del INcl String) (Write & Del INcl String) WDINS

WTOS <filename> <string test>

WDTOS <filename> <string test>

WINS <filename> <string test>

WDINS <filename> <string test>

Write (W..S) or write-and-delete (WD..S) lines, starting with the current line up to and including (WINS/WDINS) or not including (WTOS/WDTOS) the first line which satisfies the <string test>.

The <string test> has one of the following forms:

F <string>
L <string>
S (<cols>) <string>
XS (<cols>) <string>

which have the same meanings as when used alone as commands.

If query is set, the line satisfying the <string test> is typed. If this is the desired terminating line, respond Y or C; if not, respond N or Q.

If <filename> already exists, the message

File exists, rename old version or <cr>.

is typed. Enter one of:

- <cr> to replace the existing file;
- a new <filename> to be used to rename the existing file before saving the workfile;
- . (or any other invalid <filename>) to terminate the command.

When the workfile is saved, the following message is typed:

n lines written to <filename>.

If 'W..S:S' or 'WO..S:S' is specified, the lines are squished.

Toggles affecting the output of this command: AUTOBACKUP, Q, V, #.

Pointer after command completion:
WTOS : unchanged
WOTOS: current line - 1.

Examples:

```
u> P -5
--> The quick brown
    fox jumped
    over the lazy
    dog's
    back.

u> Q
    Query off.
u> WTOS FILE5 L the
        3 lines written to FILE5.

u> Q
    Query on.
u> WOTOS FILE6 F do
    dog's
    ?>
u> Y
        4 lines written to FILE6.
u> P -2
--> the line before "The quick brown"
    back.
```

X (column markers) (column markers) X

X <ncols>

A line is typed marking <ncols> columns. If <ncols> is omitted, the maximum line length is used.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
u> X 35
   ....5...10...15...20...25...30...35
u> WHAT
   (user)MYFILE
      ...
   Truncation length =  72
      ...
u> X
   ....5...10...15       ...       ...65...70..
```

. (change mode toggle) (change mode toggle) .

Change the mode of NETED from Edit-to-Input or from Input-to-Edit. The name of the mode being entered is always typed in response to this command.

When in Edit mode, all NETED commands may be entered.

When in Input mode, this command is the only NETED command recognized.

Default: If the file being edited exists: Edit mode.
If the file does not exist : Input mode.

Toggles affecting the output of this command: none.

Pointer after command completion:
Edit-to-Input: unchanged.
Input-to-Edit: at the last line entered in Input mode.

Examples:

```
u> RUN *NETED (" ")
   B7700 NETED
   Workfile is FORTRAN (fixed format)
   Truncation length =  72
   Tab character =    tab stops = 7, 10, 13, 16, ...
   Input.
u> .
   Edit.
u> .
   Input.

u> EDIT "JOB/RUNPROG"
   Workfile is JOB
   Truncation length =  72
   Tab character = \ tab stops =  3,  5,  7,  9, ...
      13 lines read from (user)JOB/RUNPROG.
   Edit.
u> P -$
--> <Top of file>
   ...
   <Bottom of file>
```

(show line numbers toggle) (show line numbers toggle)

#

Flip the show-line-numbers toggle.

When this toggle is set, commands which cause lines in the workfile to be listed will also list the line number.

When this toggle is reset, line numbers will not be listed.

Default: reset.

Toggles affecting the output of this command: none.

Pointer after command completion: unchanged.

Examples:

```
 u> P
--> this is line 17
 u> #
    Show line numbers set.
 u> P
--> 17 this is line 17
```

= (repeat last command) (repeat last command) =

=

Repeat the last command. Only the actual command portion of the last command may be re-executed. The <line#> and <toggles> are not <TOGGLES> ARE NOT retained, but may be re-entered with this command.

The only class of commands which cannot be repeated is:
commands which only move the pointer: N, <cr>, -.

Toggles affecting the output of this command: none.

Pointer after command completion: depends on command being re-executed.

Examples:

```
u> 0 P 20
   <Top of file>
   ...
--> line 19
u> =
   line 19
   ...
--> line 38
u> 101 =
   line 101
   ...
--> line 120
```

% (type at terminal) (type at terminal) %

% <string>

Type a line at the terminal. Both the '%' and the <string> are typed.

One use of this is to provide comments in a DO-file.

Another use is to quickly test the terminal and phone lines if it appears that there may be transmission problems.

Toggles affecting the output of this command: V.

Pointer after command completion: unchanged or at line <abs>.

Examples:

```
u> % This is a comment.
   % This is a comment.
```

*** A Summary of the NETED Commands by Function ***

The following terms are used in the description of NETED commands:

<col>
Column range. It has one of the following forms:

col1 -- a single column
(for string commands, the string must start in this column)

col1:ncols -- a starting column and a number of columns
(for string commands, the string must start within these columns)
(ncols = $ means maximum line length)

col1-col2 -- a range of columns
(for string commands, the string must be within these columns)
(col2 = $ means maximum line length)
(col1-$ and col1:$ are the same)

<from> - <to>
Selects one or a range of lines to be merged from a file.
'<from>-$' merges lines <from> through end-of-file.

<from> : <nlines>
Selects <nlines> lines to be merged from a file.
'<from>:$' merges lines <from> through end-of-file.
(<from>:$ and <from>-$ are the same)

<nlines>
As a separate parameter, it is the number of lines to be processed by the command. It is either a whole number, <n>, meaning process <n> lines. If <nlies> is preceded by a minus sign,'-', the pointer is returned, after processing, to its position before the command was executed.

<string>
Any string of EBCDIC characters. Trailing blanks are significant.

<string test>
Used in the 'TO String' commands. It has the same format as the F, L, S, or XS commands. For example,
DTOS S (7:3) END

Any command, except - and <cr>, may be preceded by an absolute line number. When it is, the pointer is positioned at that line before the command is executed. If there is a syntax error in the command, the pointer will be at its new position.

Any command may be preceded by one or more (comma- or blank-separated) toggles followed by a colon (:). This will change the toggle(s), for that command only, as follows:

```
        <toggle>  --  flip the toggle
      + <toggle>  --  set the toggle
      - <toggle>  --  reset the toggle
```

The general syntax of a NETED command is:

```
[ <line#> ] [<toggles> : ] <command> [ : <options> ] [ <parameters> ]
```

*** Toggles and Switches ***

Page	Command and Description
2-3	AUTOBACKUP Change existing <filename> to BACKUP/<filename> for all writes. (Default: set)
2-10	CAPS Caps lock toggle. Controls input upper/lower case translation. (Default: reset)
2-3	NOBACKUP Reset AUTOBACKUP. User has option to rename existing <filename> before all writes. (Default: reset)
2-54	PHWIDTH <nchar> Define the terminal line width for the PH and PHE commands.
2-55	POP <toggle list> Pop one or more toggles to their next previous setting.
2-57	Q Query toggle. Controls most 'TO String' commands. (Default: set)
2-62	RESET <toggle list> Reset one or more toggles.
2-66	SET <toggle list> Set one or more toggles.

Page	Command and Description
2-67	SLOW Set/reset slow print toggle. Controls printing without/with backspaces instead of carriage return.
2-77	TRANS <tbl> TRANS ? Change translate table for terminal output. (Default: 1 (non-graphic to ?))
2-78	V Verify toggle. (Default: set)
2-89	. (period) Change-mode toggle. (Edit-to-Input or Input-to-Edit)
2-90	# (number) Show-line-numbers toggle. (Default: reset)

*** Commands Which Affect Commands ***

Page	Command and Description
2-31	= (equal) Repeat the previous command.

*** Commands Which Move The Pointer ***
*** Without Changing the Workfile ***

Page	Command and Description
2-4	B Go to the bottom of the file.
2-24	F <string> FB <string> Find next (last) line containing <string> starting in column 1.
2-24	L <string> LB <string> Locate the next (last) line containing <string> anywhere in the line.
2-56	N <nlines> Move the pointer <nlines> forward or backward.
2-49	P <nlines> (<cols>) Print lines forward single-spaced.
2-49	PB <nlines> (<cols>) Print lines backward single-spaced.
2-49	PD <nlines> (<cols>) Print lines forward double-spaced.
2-53	PH <nlines> (<cols>) Print lines in hexadecimal.
2-53	PHE <nlines> (<cols>) Print lines in EBCDIC and hexadecimal.
2-56	PINS <string test> Print through and including the first line satisfying the <string test>.
2-56	PTOS <string test> Print to, but not including, the first line satisfying the <string test>.
2-24	S (<cols>) <string> SB (<cols>) <string> Scan and print next (last) line which contains <string> in the specified column range.

Page	Command and Description
2-73	T Go to the top of the file.
2-24	XS (<cols>) <string> XSB (<cols>) <string> Scan and print the next (last) line which does not contain <string> in the specified columns.
2-48	<cr> Same as 'N 1'.
2-48	- (minus) Same as 'N -1'.
2-48	<line#> Move the pointer to a specific line and print.

*** Print Commands Which Never Move The Pointer ***

Page	Command and Description
2-26	FA <string> FBA <string> Find all following (preceding) lines which contain <string> starting in column 1.
2-26	LA <string> LBA <string> Locate all following (preceding) lines containing <string> anywhere in the line.
2-37	LPL <page length> List page lengths.
2-51	PA (<cols>) Print all lines single-spaced.
2-51	PDA (<cols>) Print all lines double-spaced.
2-60	RA <nlines> (RAnge). Print lines around the current line.
2-26	SA (<cols>) <string> SBA (<cols>) <string> Scan and print all following (preceding) lines which contain <string> in the specified column range.
2-26	XSA (<cols>) <string> XSBA (<cols>) <string> Scan and print all following (preceding) lines which do not contain <string> in the specified columns.

*** Commands Which Alter Lines ***

Page	Command and Description
2-2	A <string> Append <string> to the end of the line.
2-2	AL <string> Add <string> to the start of the line.
2-2	AR <string> Add <string> to the end of the line. (Same as A.)
2-5	BR <col> Break a line at column <col>.
2-6	C /<string1>/<string2>/ (<cols>) *<ntimes> <nlines> <global> <?> Change <string1> to <string2> forward.
2-6	CB /<string1>/<string2>/ (<cols>) *<ntimes> <nlines> <global> <?> Change <string1> to <string2> backward.
2-12	CO <nlines> AT <line#> CO <nlines> +/- <mlines> Copy lines, retaining the original.
2-15	D <nlines> Delete lines.
2-15	DB <nlines> Delete lines backward.
2-18	DINS <string test> Delete through and including first line satisfying the <string test>.
2-17	DTOP Delete from <Top of file> to pointer.
2-18	DTOS <string test> Delete to, but not including, first line satisfying the <string test>.
2-19	DUP <ntimes> Duplicate the current line <ntimes> times.

Page	Command and Description
2-28	FIX <nlines> FIX +/- Modify a line.
2-32	FOLD (<cols>) <nlines> Change lower case to upper case.
2-36	I <string> Insert a line.
2-5	K <ncols> <nlines> KL <ncols> <nlines> Keep left-most columns.
2-5	KR <ncols> <nlines> Keep right-most columns.
2-46	MO <nlines> TO <line#> MO <nlines> +/- <mlines> Move lines. Do not retain the original.
2-59	R <string> Replace a line.
2-6	RC (<cols>) *<ntimes> <nlines> <global> <?> Repeat last change forward.
2-6	RCB (<cols>) *<ntimes> <nlines> <global> <?> Repeat last change backward.
2-68	SR <ncols> <nlines> Shift lines to the Right.
2-76	TL <ncols> <nlines> Truncate columns on the left.
2-76	TR <ncols> <nlines> Truncate columns on the right.
2-6	UC (<cols>) *<ntimes> <nlines> <global> <?> Undo last change forward.
2-6	UCB (<cols>) *<ntimes> <nlines> <global> <?> Undo last change backward.
2-32	UNFOLD (<cols>) <nlines> Change upper case letters to lower case.

*** Commands Which Operate in Hexadecimal ***

Page	Command and Description
2-53	PH <nlines> (<cols>) Print lines in hexadecimal.
2-53	PHE <nlines> (<cols>) Print lines in EBCDIC and hexadecimal.

*** File Manipulation Commands ***

Page	Command and Description
2-20	EDIT "<filename>,<filekind>,<length>" Edit a new workfile.
2-44	M <filename> M <filename> <from> - <to> M <filename> <from> : <nlines> Merge all (or part) of a file.
2-61	REC <recovery number> Edit a recovery file.
2-63	SAVE <filename> Save the workfile and terminate NETED.
2-65	SEQ <base> + <incr> SEQ ? Define or interrogate base and increment for writing files.
2-74	TEMP <temporary number> Edit a temporary file.
2-80	W <filename> Write entire workfile to file <filename>.
2-86	WDINS <filename> <string test> Write-to-a-file-and-delete lines through and not including the first line satisfying the <string test>.
2-83	WDL <nlines> <filename> Write lines to a file and delete the lines.

Page	Command and Description
2-85	WDTOP <filename> Write-and-delete lines from <Top of file> to pointer.
2-86	WDTOS <filename> <string test> Write-to-a-file-and-delete lines to, but not including, the first line satisfying the <string test>.
2-86	WINS <filename> <string test> Write to a file through and including the first line satisfying the <string test>.
2-84	WL <nlines> <filename> Write lines to a file.
2-85	WTOP <filename> Write lines from <Top of file> to pointer.
2-86	WTOS <filename> <string test> Write to a file to, but not including, the first line satisfying the <string test>.

*** Commands Which Terminate NETED ***

Page	Command and Description
2-58	QUIT Terminate NETED without saving the workfile.
2-63	SAVE <filename> Save the workfile and terminate NETED.

*** Specialized Commands ***

Page	Command and Description
2-22	ENTER ENTER <nsecs> Input lines from an auxilliary storage device (such as a Techtran 9125 floppy disk on an Omron CRT).
2-71	STORE STORE <nlines> Write lines to an auxilliary storage device (such as a Techtran 9125 floppy disk on an Omron CRT).

*** Special listing commands ***

Page	Command and Description

<<< possible future commands >>>

*** Text/Word Processing Related Commands ***

Page	Command and Description
2-11	CENTER <nlines> Center lines within LM-RM margins.
2-14	COUNTS List the workfile statistics.
2-37	IN <ncols> Indent/undent from left margin.
2-11	LEFT <nlines> Left-justify lines within LM-RM margins.
2-39	LEN <nlines> <long> List line lengths, optionally flagging long lines.
2-40	LM <lm> LM ? Set the left margin for CENTER, LEFT, RIGHT, FILL, FILLJ.
2-42	LPL <page length> List page lengths.
2-11	RIGHT <nlines> Right-justify lines within LM-RM margins.
2-40	RM <rm> RM ? Set the right margin for CENTER, LEFT, RIGHT, FILL, FILLJ.

*** Miscellaneous Commands ***

Page	Command and Description
2-9	CANDE Suspend NETED to execute CANDE commands.
2-34	H or HELP Type a list of the current NETED commands.
2-41	LN Type the line number of the current line.
2-52	PFN <filename> PFN ? Change the name of the workfile.
2-69	STAB tabch tab1, tab2, ... STAB tabch <base> + <incr> STAB STAB ? Set or interrogate tab character and tab stops.
2-72	STR List the current string definitions.
2-75	TI Show the elapsed, process and I/O times and the process time remaining.
2-79	VERSION List the NETED compile date/time and recent modifications.
2-82	WHAT < long form -V:WHAT < short form List the workfile attributes.
2-88	X <ncols> Type a line of column markers.
2-92	% <string> Type <string> at the terminal.

Appendix A

*** Executing NETED ***

To execute NETED:

RUN *NETED ("<filename>,<filekind>,<length>")

where

<filename> - the file to be edited. One of:
- an existing file
- a non-existent file
  (to create a new file)
- REC <recovery number>
  (to re-edit and remove a recovery file)
- TEMP <temporary number>
  (to re-edit and remove a temporary file)

If <filename> is preceded by a minus sign, '-', or a plus sign, '+', the short form of the initial NETED messages will be given. The '+' will also set MAXPROCTIME to 300 seconds. (See the EDIT command.)

<filekind> - type of file, optional unless creating a file
One of:

ALGOL	or	A	- Algol
BASIC	or	BA	- Basic
BINDER	or	BI	- Binder
CDATA	or	CD	- Cdata
CSEQ	or	CS	- Cseq
COBOL	or	C	- Cobol
DASDL	or	DAS	- Dasdl
DATA	or	DA	- Data
DCALGOL	or	DC	- Dcalgol
DMALGOL	or	DM	- Dmalgol
ESPOL	or	ES	- Espol
FORTRAN	or	F	- Fortran (fixed format)
FREE	or	FF	- Fortran (free format)
JOB	or	J	- Job
NDL			- Ndl
NEWP			- Newp
PASCAL	or	P	- Pascal (generates JOVIALSYMBOL until MCP supports PASCALSYMBOL)
PLI	or	PL	- PL/I
SEQ	or	SE	- Seqdata

(if omitted, FORTRAN is assumed for a new file)

<length> - optional maximum line length
(if omitted, the default for the <filekind> is used)

Appendix B

*** NETED Messages ***

Listed in this Appendix are all the NETED messages, their causes and fixes (if appropriate) and the command or commands which issue the message. If a cause-and-fix is not listed, the message is either self-explanatory or is a message which should never occur unless there are NETED or system problems. A command is not listed if the message is general or may be caused by a great many commands.

Message	Cause and Fix	Command
Another program has exclusive access to this file.		
Attempt to read illegal filekind no lines read from <filename>		EDIT,NETED
AUTOBACKUP already set.		AUTOBACKUP
AUTOBACKUP set.		AUTOBACKUP
Backed up.	AUTOBACKUP was set and the existing <filename> was changed to BACKUP/<filename> before the new file was written.	all write commands
base + incr = <base> + <incr>	Response from 'SEQ ?'.	SEQ
<Bottom of file>		
Break-on-output, command aborted.	User pressed the <break> key to terminate printing.	
B7700 NETED <version>		header
Cannot add to top-or bottom-of-file.	Self-explanatory. Move the pointer to a line in the workfile and re-issue the command.	A/AR, AL
Cannot center top- or bottom-of-file.	Self-explanatory. Move the pointer to a line in the workfile and re-issue the command.	CENTER

Message	Cause and Fix	Command
Cannot copy top- or bottom-of-file.	Self-explanatory. Move the pointer to a line in the workfile and re-issue the command.	CO
Cannot duplicate top- or bottom-of-file.	Self-explanatory. Move the pointer to a line in the workfile and re-issue the command.	DUP
Cannot left-justify top- or bottom-of-file.	Self-explanatory. Move the pointer to a line in the workfile and re-issue the command.	LEFT
Cannot move top- or bottom-of-file.	Self-explanatory. Move the pointer to a line in the workfile and re-issue the command.	MO
Cannot replace top- of bottom-of-file.	Self-explanatory.	R
Cannot right-justify top- or bottom-of-file.	Self-explanatory. Move the pointer to a line in the workfile and re-issue the command.	RIGHT
<existing filename> changed to <new filename>.		all write commands
Column greater than max. line length.		(<cols>)
Column range error.		(<cols>)
Column range less than text length.		(<cols>)
Column range syntax error.		S, SA, SB, SBA, XS, XSA, XSA, XSBA
Command aborted.	User response to query was N or Q.	query mode
Copy syntax error.	Self-explanatory.	CO
Currently-defined strings:	First line of output for STR command.	STR

Message	Cause and Fix	Command
Deleted to bottom.		D, WDL
Directives ABEND reset.		ABEND
Directives ABEND set.		ABEND
DIRectives LIST reset.		DIRLIST
DIRectives LIST set.		DIRLIST
Directives STEP reset.		STEP
Directives STEP set.		STEP
Disk available error - command aborted.		
Disk blocksize error - command aborted.		
Disk filekind error - command aborted.		
Disk filetype error - command aborted.		
Disk maxrecsize error - command aborted.		
Disk minrecsize error - command aborted.		
Disk open error - command aborted.		
Disk present error - command aborted.		
Disk read data/format error.		
Disk read parity error.		
Disk read security error.		
Disk read time limit error.		
Disk resident error - command aborted.		
Disk security guard error - command aborted.		
Disk security type error - command aborted.		
Disk security use error - command aborted.		
Disk write data/format error.		
Disk write parity error.		
Disk write security error.		
Disk write time limit error.		

Message	Cause and Fix	Command
Due to an error while creating the Workfile either a new Workfile must be established or NETED abandoned. Please enter: EDIT "<filename>,<filekind>,<length>" or QUIT.	The status of the workfile is undefined at this point. In order to continue, a new file must be edited. All other commands will be ignored.	EDIT
Edit.	NETED is now in Edit mode. All NETED commands may be entered.	., ENTER
End of file encountered.		
Enter valid EDIT, or QUIT command.	An EDIT command or RUN statement with invalid syntax has been encountered. Since the workfile has been reset, only a valid EDIT, or QUIT command will be accepted.	EDIT
EOF encountered on disk read.		
EOF encountered on disk write.		
Error - end of file encountered while skipping to first merge line, no lines read from <filename>.		
Error in optional parameters.		C
Error - tabs must be in ascending order.		
Escape character error.	<esc> may only be followed by <esc>, 0, 1, 2, 3, 4.	All string commands
>ET=<et> PT=<pt> IO=<io>		QUIT, SAVE, TI
File assignment denied.		
File assignment postponed.		

Message	Cause and Fix	Command
File exists, rename old version or <cr>.	A file already exists with the <filename> you are trying to use. Enter one of: - <cr> to replace the existing file - a new <filename> to be used to rename the exisginf file before writing the new file - . (or any invalid <filename>) to terminate the command.	all write commands
Filekind not supported by NETED.		EDIT,NETED
File not present.		
Filename required.		M, all writes
File squished.	':S' option was specified on a write.	all write commands
File title error - command aborted.		
File too big.	The file being edited/merged has too many lines for NETED. Break up the file or use the CANDE EDITOR.	EDIT, M, NETED
Getstatus errer # <n>	Written to CONSOL and TASKFILE. Please forward this printout, and a description of what you were doing, to User Services.	
Illegal argument.		CANDE, D, K, KL, KR, SR, TL, TR, REC, TEMP
Illegal file use.		
Illegal optional parameter.		C
Illegal parameter.		S, SA, SB, SBA, XS, XSA, XSB, XSBA

Message	Cause and Fix	Command
Individual HELP not available for <command>	Self-explanatory. Use HELP 13 for a list of commands having individual HELPs.	HELP
Input.	NETED is reading from an external storage device. NETED will return to Edit mode after a 'stop read' has been read or there is no input for 5 seconds.	ENTER
Input.	NETED is now in Input mode. The only NETED command recognized is '.'.	.
Input caps lock reset.		CAPS
Input caps lock set.		CAPS
Input file maxrecsize too big for filekind.		EDIT,NETED
Input filekind is <filekind>, interpreted as CDATA tr=<n> <m> lines	Input filekind is not recognized by NETED. Since UNITS=CHARACTERS, CDATA is used.	EDIT, RUN
Input filekind is <filekind>, interpreted as DATA tr=<n> <m> lines	Input filekind is not recognized by NETED. Since UNITS=WORDS, DATA is used.	EDIT, RUN
Invalid DESTNAME.		DESTNAME
Invalid - must be >= left margin (<lm>).		RM
Invalid - must be <= maximum line length (<length>).		RM
Invalid - must be <= right margin (<rm>).		LM
Invalid - must be positive.		LM, RM
Invalid filekind: <text>		EDIT,NETED
Invalid - no old string in buffer.		S, SA, SB, SBA, XS, XSA, XSB, XSBA
Invalid - no old tabs.		STAB
Invalid number.		

Message	Cause and Fix	Command
--Invalid response, must be one of Y,C,N,Q	Y - Accept line C - Accept line, continue command without ? N - Do not accept line, continue with command Q - Quit command.	C
--Invalid response, must be one of Y,C,N,Q	Y or C - Continue N or Q - Abort.	xINS, xTOS F/L/S/XS/ etc.
--Invalid response, must be one of Y,C,N,Q	Y or C - Continue VERSION N or Q - Abort VERSION.	VERSION
Invalid toggle: <name>		POP, RESET, SET
Invalid - too long.		IN
Invalid - too small.		IN
Invalid translate table #.		TRANS
I/O descriptor = <descriptor in hex>	Written to TASKFILE.	
I/O error - Invalid # of characters returned by I/O subsystem.	Written to CONSOL and TASKFILE. Please forward this printout, and a description of what you were doing, to User Services.	
Line <n>		LN
Line duplicated <n> times.		DUP
<n> lines, <m> characters Shortest=<s> ,longest=<l>		COUNTS
<n> lines copied, inserted after line <m>.		CO
<n> lines copied, inserted after <Top of file>		CO
<n> lines copied, inserted at <Bottom of file>		CO
<n> lines deleted.		DINS, DTOS
<n> lines entered.		ENTER

Message	Cause and Fix	Command
<n> lines moved, inserted after line <m>.		MO
<n> lines moved, inserted after line <m> (updated).		MO
<n> lines moved, inserted after <Top of file>		MO
<n> lines moved, inserted at <Bottom of file>		MO
<n> lines read from <filename>.		EDIT, M, NETED
<n> lines stored.		STORE
<n> lines truncated.		ENTER
<n> lines written to <filename>.		all write commands
Line too long to center.	Line does not fit in within [LM-RM].	CENTER
Line too long to left-justify.	Line does not fit in within [LM-RM].	LEFT
Line too long to right-justify.	Line does not fit in within [LM-RM].	RIGHT
Maximum line length = <length>		EDIT,NETED long form
Max. line length for CDATA is 255.		EDIT,NETED
Max. line length for CSEQDATA is 246.		EDIT,NETED
Max. line length for DATA is 252.		EDIT,NETED
Merge range error.	An invalid line range was specified. Re-enter the command with a valid line range.	M
More?		VERSION
Move syntax error.		MO

Message	Cause and Fix	Command
Must specify column range.		S, SA, SB, SBA, XS, XSA, XSB, XBSA
Must supply a valid F, L, S, or XS command.		xINS, xTOS
Must supply at least <filename>		EDIT
Must supply <filename> followed by a valid F, L, S, or XS command.		
Must supply toggles		POP, RESET, SET
NETED fault # <n> <stack history>	Written to CONSOL and TASKFILE. Please forward this printout, and a description of what you were doing, to User Services.	
NETED recovery file(s) present:	First line of message listing recovery files. Use 'REC <n>' to edit a recovery file.	EDIT, REC, TEMP
NETED suspended for <n> minutes (suicide at <hh:mm>!). To resume - ? <mixno> HI	You are back under CANDE and any CANDE commands may be entered. Unless NETED is re-awakened before the specified time, it will commit suicide.	CANDE
NETED temporary file(s) present:	First line of message listing temporary files. Use 'TEMP <n>' to edit a temporary file.	EDIT, REC, TEMP
NETED <version> commands:		HELP, H
NETED Version <version>, compiled on <date> @ <time>		VERSION VER
No available tape drive.		
NOBACKUP already set.		NOBACKUP
NOBACKUP set.		NORACKUP

Message	Cause and Fix	Command
No last command.	There is no command to repeat.	=
No lines above pointer.		DTOP, WDTOP, WTOP
No lines deleted.		WDINS, WDL, WDTOP, WDTOS
No old Scan columns.		S, SA, SB, SBA, XS, XSA, XSB, XSBA
No old text in buffer.		A/AR, AL
No tabs set.		STAB
Not a NETED command: <text>	NETED could not decipher <text> as a valid command.	
Not found.		
Nothing changed.		C, RC, UC
Old version retained as = <filename>.		all write commands
Option syntax error.		commands with options
<n> PT seconds remain - please save Workfile.	There may not be enough time left to do much more than save the workfile. Do so and re-execute NETED, perhaps with a higher MAXPROCTIME.	
Query reset.		Q
Query set.		Q
Range overlap error, no lines moved.		MO
Recovery file created, title=<filename>.	Written to terminal and TASKFILE.	

Message	Cause and Fix	Command
Requested line length too big for this filekind - ignored.	Self-explanatory.	EDIT,NETED
Recovery file removed.		EDIT,NETED REC
Required disk pack not mounted.		
>Remaining PT=<pt>.	Line 2 of TI output. Also printed when remaining time is getting low.	
>RUNNING <mixno>	Part of the long form of the NETED prologue.	header
Security guard file info lost.	Written to terminal and TASKFILE.	
Security violation.		
SEQ syntax error.		SEQ
Sequence numbers not valid for this filekind.	Sequence number may not be specified for BASIC, CDATA or DATA files.	SEQ
Show line numbers reset.		*
Show line numbers set.		#
Slow printing reset.		SLOW
Slow printing set.		SLOW
STAB syntax error.		STAB
Suspended time limit exceeded.	Written to TASKFILE when the NETED suspension time has elapsed and the user is busy or disconnected.	CANDE
Syntax error.		
Tab character = <tabch> tab stops = <tabs>		STAB EDIT,NETED long form
Tab character changed.		STAB

Message	Cause and Fix	Command
Tabs off.		STAB
Tabs restored.		STAB
Terminal width = nnn	Self-explanatory.	PHWIDTH
Terminal width must be > 0.	Self-explanatory.	PHWIDTH
Temporary file removed.		EDIT, NETED TEMP
The following lines were not centered: <line numbers>	Self-explanatory.	CENTER
The following lines were not left-justified: <line numbers>	Self-explanatory.	LEFT
The following lines were not right-justified: <line numbers>	Self-explanatory.	RIGHT
Timelimit exceeded.		
Toggle prefix error.	Something is wrong in the toggle prefix of the last command entered.	
Too many options (<n> max)	Self-explanatory.	commands with options
<Top of file>	Indicated the pointer is at the top of the file.	
Translate table menu:	First line of output for 'TRANS ?'.	TRANS
<text> truncated.	The line just entered or read was too long for the maximum line length.	A/AR, AL, C, I, input mode
TTY data/format error - command aborted.		
TTY I/O parity error - command aborted.		

Message	Cause and Fix	Command
TTY Output translation = EBCDIC to ASCII.	Indicates that all lines will be typed as is, except the EOT character.	TRANS
TTY Output translation = non-graphic to blank.	Indicates that all lines will be typed with non-printing characters changed to blanks.	TRANS
TTY Output translation = non-graphic to ?.	Indicates that all lines will be typed with non-printing characters changed to question mark '?'.	TRANS
Unknown disk error, attribute # <n> - command aborted.		
Unknown disk read error.		
Unknown disk write error.		
Unknown file-available error.		
Unknown TTY I/O error.		
Unmatched delimiters.		C, EDIT
Unmatched geneology.		
Unmatched SERIALNO.		
Verify reset.		V
Verify set.		V
Warning - requested line length differs from default for this filekind.		EDIT,NETED
Warning - workfile and input filekinds differ.		EDIT,NETED
Warning - workfile and input maxrecsizes differ.		EDIT,NETED

Message	Cause and Fix	Command
Workfile full.	NETED supports only a finite number of lines, dependent upon the maximum line length. This limit has been reached. To add more lines, either delete some lines first or break up the workfile into 2 or more parts and edit each part separately.	CO, DUP
Workfile not saved.		EDIT, QUIT
><mixno> (<version>)	Part of the short form of the NETED prologue.	header

INITIAL DISTRIBUTION

COPIES:

12 DIRECTOR
DEFENSE DOCUMENTATION CENTER (TIMA)
CAMERON STATION
ALEXANDRIA, VIRGINIA 23314

CENTER DISTRIBUTION

COPIES:

1	18/1809	GLEISSNER, G. H.
1	1804	AVRUNIN, L.
2	1809.3	HARRIS, D.
1	182	CAMARA, A. W.
1	184	SCHOT, J. W.
1	185	CORIN, T.
1	187	ZUBKOFF, M. J.
1	189	GRAY, G. R.
1	189.1	HIBBERT, D.
1	189.2	HAYDEN, H. P.
1	189.3	MORRIS, JEAN
20	1892.1	STRICKLAND, J. D.
10	1892.2	SOMMER, D. V.
1	1892.3	MINOR, L. R.
1	1894	SEALS, W.
1	1896	GLOVER, A.
1	1896.2	DENNIS, L.
10	2731	ZIVI, E. L.
1	522	LIBRARY, CARDEROCK
1	522.2	LIBRARY, ANNAPOLIS

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.


DA1
FILM